



# GUIDE D'INTRODUCTION AU VM/CMS

*Numéro de document : ZZ01-0090-14*

*16 septembre 1996*

Service Général d'Informatique

Université de Liège  
Campus universitaire du Sart Tilman  
Bâtiment B26 - Parking 32  
Sart Tilman - 4000 Liège  
Tél. 04/366.49.04



# Avant-propos

Etant donné la grande variété des logiciels en service, le SEGI a voulu mettre à la disposition des utilisateurs une brochure décrivant de manière simplifiée l'utilisation de ces différents produits sur ses ordinateurs.

Quant au fond, cette brochure est conçue comme un guide, permettant à chacun de trouver rapidement ce qui est disponible sur le système d'exploitation VM, ne s'attachant qu'aux fonctions les plus utilisées et décrivant les modalités d'accès aux ordinateurs. Pour certains détails techniques, il faudra donc recourir aux publications propres à chaque logiciel (publications disponibles au bureau de documentation du SEGI).

Quant à la forme, elle se veut surtout pratique et sujette à recevoir régulièrement (normalement une fois par an à l'occasion de la rentrée académique) les modifications requises par l'évolution de l'outil informatique. C'est pourquoi, cette brochure est encodée et mise en forme par le logiciel de composition de documents (DCF) disponible au SEGI.

On notera que les ajouts et modifications par rapport à l'édition antérieure sont signalés par un trait vertical dans la marge.



# Table des matières

<b>1.0 Introduction</b>	<b>1</b>	<b>8.0 Procédures d'exploitation</b>	<b>85</b>
<b>2.0 Contacts utiles.</b>	<b>3</b>	8.1 Tableau des limitations par classe (travaux batch)	85
<b>3.0 Concepts du VM.</b>	<b>5</b>	8.2 Utilisation des périphériques réels	86
<b>4.0 Le CP</b>	<b>7</b>	8.3 Accès via le réseau commuté	86
4.1 Démarrer, arrêter ou déconnecter une machine virtuelle	7	8.4 Procédures de back-up système	86
4.2 Définir l'architecture de la machine virtuelle	8	8.5 Conservation des fichiers dans un spool VM	86
4.3 Définir le hardware de la machine virtuelle	8	<b>9.0 Logiciels d'applications</b>	<b>89</b>
4.4 Gestion des unit records	9	9.1 SAS	89
4.5 IPL: charger le système CMS	13	9.2 La NAG FORTRAN Library.	92
4.6 Questionner le CP sur l'état de la machine virtuelle.	13	9.3 Le système BMDP.	93
4.7 Facilités particulières du CP	14	9.4 Le logiciel GLIM	94
<b>5.0 Le CMS</b>	<b>17</b>	9.5 Le logiciel TSP	94
5.1 Utilisation d'un minidisque en CMS	17	9.6 Le logiciel STATPACK	95
5.2 L'édition en CMS	25	9.7 Applications graphiques.	95
5.3 Dialogues entre machines virtuelles.	36	9.8 Le logiciel de recherche documentaire: Search Manager	98
5.4 Les interpréteurs de commandes.	37	9.9 Les bases de données relationnelles: SQL	98
5.5 La commande HELP.	43	9.10 PUBLISHING: un environnement complet basé sur les logiciels DCF pour la partie texte et GDDM pour la partie graphique.	98
5.6 Modes de fonctionnement et procédures d'interruption	45	9.11 La PUBLIB.	102
5.7 Programmation sous CMS	45	9.12 ISPF et ISPF/PDF	102
5.8 Sauvegarde de l'environnement VM.	53	9.13 Logiciel de calcul symbolique: REDUCE.	107
5.9 CMS BATCH	53	<b>10.0 Aspects administratifs et comptables</b>	<b>109</b>
5.10 Impression en format A4: imprimante 3812	60	10.1 Structure et fonctionnement des comptes	109
5.11 Utilisation des bandes et des cassettes sous VM/CMS	61	10.2 Ouverture des comptes et des machines virtuelles	109
5.12 Space Manager VM	67	10.3 Imputation des consommations sur les comptes	110
5.13 Modification des caractéristiques d'une machine virtuelle	70	10.4 Notification des coûts aux utilisateurs	110
<b>6.0 La sécurité du système et ACF2</b>	<b>73</b>	10.5 Gestion des comptes par la commande CREDIT	110
6.1 Généralités	73	10.6 Services particuliers facturés aux utilisateurs	111
6.2 Gestion des mots de passe d'une machine virtuelle	73	<b>11.0 Bibliographie</b>	<b>113</b>
6.3 Gestion et protection des minidisques	74	11.1 IBM	113
6.4 Quelques protections supplémentaires permises par ACF2	74	11.2 SEGI	114
6.5 Quelques messages ACF2	75	11.3 SAS	114
<b>7.0 La communication au SEGI</b>	<b>77</b>	11.4 DISSPLA	115
7.1 Introduction	77	11.5 Divers	115
7.2 Réseau Interactif VTAM/SNA	77	<b>Annexe A. Formulaires</b>	<b>117</b>
7.3 Réseau networking	78	A.1 formulaire 1	117
7.4 Le courrier électronique	80	A.2 formulaire 2	117
7.5 TCP/IP	81	A.3 formulaire 3	118
7.6 Moyens matériels	81	A.4 formulaire 4	118
		A.5 formulaire 5	119
		A.6 formulaire 6	119

**Annexe B. Les jeux de caractères employés au SEGI ..... 121**

**Index ..... 127**

# 1.0 Introduction

La présente brochure a pour objet de fournir les informations techniques nécessaires à l'utilisation de l'ordinateur central "tous usages" du SEGI, à savoir actuellement:

un superordinateur IBM ES/9000 (modèle 9121-621) avec les caractéristiques suivantes:

- deux processeurs scalaires d'une puissance de 29 Mips chacun;
- un processeur vectoriel d'une puissance de 30 Mflops (sur le LINPACK DP test);
- une mémoire centrale de 512 Mbytes;
- une mémoire d'arrière-plan de 1024 Mbytes;
- des disques IBM 3390 d'une capacité globale de 112,08 Gbytes;
- cinq dérouleurs de bandes MEMOREX, compatibles IBM 3420, pour bandes magnétiques à 9 pistes (densité 1600/6250 BPI);
- quatre unités à cassettes IBM 3480 (densité 38.000 BPI);
- une imprimante laser SIEMENS compatible IBM 3800 (100 pages/minute);

- une table traçante CALCOMP 4 couleurs (25 cm/sec);
- des ordinateurs frontaux et contrôleurs de télécommunication pour gérer les liaisons de et vers l'extérieur.

Cet ordinateur fonctionne sous le contrôle du système d'exploitation VM ("VIRTUAL MACHINE"), software de base performant, bien adapté aux travaux interactifs (tout en admettant des traitements batch), et d'une relative simplicité d'emploi.

Dans le parc très diversifié des équipements informatiques installés dans l'Université, le superordinateur du SEGI constitue une machine centrale à la disposition de l'ensemble de la communauté; à ce titre, elle réunit tous les langages et logiciels d'application dignes d'intérêt pour la toute grande majorité des utilisateurs. Des logiciels de réseau sont par ailleurs implantés sur cette machine pour assurer la communication avec d'autres systèmes, qu'ils soient internes ou externes à l'Université; les modes d'utilisation des réseaux sont détaillés au chapitre «La communication au SEGI» à la page 77.

Remarquons enfin que ce guide est directement orienté vers l'emploi du système VM propre au superordinateur du SEGI; il ne concerne pas d'autres systèmes VM pouvant exister dans l'environnement universitaire et présentant chacun leurs particularités.



## 2.0 Contacts utiles.

On trouvera ci-après la liste des personnes et bureaux du SEGI en liaison directe avec les utilisateurs:

1. **Help-Desk** *tél.: 366.49.49/50; e-mail: helpdesk@segi.ulg.ac.be*  
Traite les problèmes strictement liés au fonctionnement des équipements centraux du SEGI (ordinateurs, périphériques, ...) ainsi que les problèmes rencontrés lors de l'utilisation du réseau ULg. Il peut également fournir des renseignements en matière de dépannage de matériel.
2. **Contrôle des opérations** (Bur: 0/14) : *tél. 366.49.59; e-mail : control@segi.ulg.ac.be.*  
Prend en charge tout problème rencontré lors de l'exploitation, au sens large, des équipements et des logiciels, ainsi que les activités suivantes:
  - a. planification des travaux spéciaux;
  - b. organisation de la restitution des résultats et service camionnette.La livraison des consommables (listings, cassettes, disquettes,...) est assurée par le personnel de la salle des machines *tél.: 366.49.50.*
3. **Service micro-informatique** (Bur.0/27) : *Mr Radoux (tél. 366.49.40).*
4. **Relations extérieures, Gestion des comptes et des crédits** (Bur.0/15): *Mr. Petit (tél. 366.49.13)*  
Toute question sur les modalités administratives et financières d'utilisation du SEGI peut lui être soumise: obtention de crédits, ouverture de comptes courants et de machines virtuelles, attribution d'espace sur disques...
5. **Documentation** (Bur.0/35): *Mme Muller (tél. 366.49.04)*  
Vend des brochures des constructeurs ou producteurs de logiciels, ainsi que des notices publiées par le SEGI, mais peut aussi prêter certains documents à des fins de photocopie.
6. **Secrétariat général - accueil** (Bur. 0/35): *Mme Muller (tél. 366.49.04)*  
Son numéro téléphonique peut être formé par les interlocuteurs n'obtenant pas de réponse de la personne appelée. Il peut réceptionner les messages et/ou transmettre la communication à une personne susceptible de répondre à l'appel.
7. **Secrétariat de direction** (Bur.0/19): *Mme Hansen (tél. 366.49.09/10)*



## 3.0 Concepts du VM.

Le VM est un système qui permet à plusieurs utilisateurs d'un même ordinateur de travailler comme s'ils disposaient chacun de leur propre machine, appelée ci-après machine virtuelle (VM = virtual machine). Un programme de contrôle, appelé CP (control program), gère l'ordinateur réel et effectue toutes les simulations et opérations nécessaires de telle manière que chaque machine virtuelle réagisse et travaille comme le ferait une machine réelle. Chaque propriétaire d'une machine virtuelle doit donc donner au CP les caractéristiques du hardware virtuel de son ordinateur. Cela fait, il peut charger sa machine virtuelle avec le software (OS, DOS, CMS, un autre VM, etc...) sous lequel il désire travailler.

Sur le superordinateur du SEGI, c'est la version VM/ESA du VM qui est installée. Le terme ESA (*Enterprise System Architecture*) désigne une nouvelle architecture de machine qui supporte l'adressage bimodal (24 ou 31 bits) et les architectures de machines virtuelles XC (*eXtended Configuration*), XA (*eXtended Architecture*) et 370.

Dans l'architecture 370, une adresse mémoire comprend 24 bits, ce qui limite l'espace adressable à 16 Mbytes. Dans les architectures XA et XC, une adresse mémoire comprend 31 bits, ce qui permet d'adresser jusqu'à 2 Gbytes. La taille des machines virtuelles est cependant limitée à 2047 Mbytes par le CMS.

**En pratique**, au SEGI, les machines virtuelles des utilisateurs sont pré-définies vis-à-vis du CP comme indiqué ci-dessous :

- architecture XA,
- 4 MB de mémoire centrale,
- 1 disque de travail,
- des disques "système" protégés en écriture et sur lesquels sont situés les logiciels publics,
- 1 lecteur de cartes (appelé reader),
- 1 perforateur de cartes (appelé punch),
- 1 imprimante (appelée printer),
- 1 console.

Chaque utilisateur peut évidemment modifier la définition de sa machine virtuelle (c'est-à-dire son hardware) pour autant qu'il reste dans les limites prévues par les procédures d'exploitation appliquées au SEGI.

A ce stade, deux remarques importantes s'imposent:

1. Les disques affectés à une machine virtuelle sont communément appelés minidisques, car ils n'occupent généralement qu'une petite portion d'un disque réel. Un disque réel est donc découpé en plusieurs minidisques appartenant à différentes machines virtuelles.
2. Au SEGI, les "unit records" de chaque machine virtuelle, c'est-à-dire le reader, le punch et le printer sont toujours virtuels. Il en résulte donc qu'une machine virtuelle ne saurait jamais lire des cartes réelles, en perforer ou imprimer des listings en utilisant ses propres "unit records": les punches et les listings sont envoyés sur disque dans un pot commun appelé SPOOL. De même, le reader ne permet de lire que des images de cartes situées dans le SPOOL. C'est le CP qui gère ce SPOOL, lequel est commun à toutes les machines virtuelles, et sert d'ailleurs de moyen de communication entre elles. Les "unit records" réels (au SEGI, uniquement des imprimantes) étant sous le contrôle du CP, la technique consistera à envoyer les listings dans le SPOOL via le printer de la machine virtuelle et à laisser le soin au CP de les y extraire pour les imprimer. Les différentes manipulations sur des fichiers "spoolés" sont décrites en détail dans le chapitre relatif au CP.

Au SEGI, tout utilisateur VM charge sa machine virtuelle avec un système conversationnel appelé CMS (Conversational Monitor System) qui permet d'effectuer la plupart des travaux informatiques. Le système CMS a été préféré à tout autre système d'exploitation, car il est spécialement conçu pour opérer dans une machine virtuelle avec un niveau de performance optimum.

Dans cette brochure consacrée à l'utilisation du VM au SEGI, vous trouverez donc, d'une part, les informations utiles pour converser avec le CP afin de définir et gérer votre machine virtuelle et, d'autre part, les informations nécessaires pour exécuter vos travaux informatiques sous le contrôle du système CMS. Néanmoins, ce guide étant introductif, les commandes qui y sont décrites ont vu leur syntaxe volontairement simplifiée.



Dès qu'un terminal est connecté au VM, l'utilisateur peut dialoguer avec le CP pour lui transmettre ses desiderata. Ceci est effectué à l'aide d'un jeu de commandes CP qui peuvent se subdiviser en six groupes.

1. des commandes pour démarrer, arrêter ou déconnecter une machine virtuelle,
2. une commande pour définir l'architecture de la machine virtuelle,
3. des commandes pour définir le hardware de la machine virtuelle,
4. des commandes pour gérer le SPOOL,
5. une commande pour charger un software de base (le CMS en l'occurrence),
6. des commandes pour bénéficier de quelques facilités particulières propres au CP.

### 4.1 Démarrer, arrêter ou déconnecter une machine virtuelle

#### 4.1.1 Démarrer une machine virtuelle (ou, dans le jargon, 'faire LOGON')

Dès qu'un terminal est opérationnel, il affiche le tableau suivant (que l'on appelle aussi "mire VTAM") qui reprend la liste des ordinateurs disponibles:

```
VM/VTAM V3
  POUR ATTEINDRE      ENTRER
VM BLIULG11(3090-SEGI) VM1 userid
  ...                ...
```

Dans cette liste on retrouve le nom donné par le SEGI à chaque ordinateur disponible (par exemple BLIULG11 est le nom donné au superordinateur installé au SEGI) et la commande à entrer pour y accéder (par exemple VM1). 'userid' représente le nom de la machine virtuelle que l'on veut démarrer sur l'ordinateur sélectionné. Une fois identifié par le système, le "userid" sera invité à introduire un mot de passe (voir «Gestion des mots de passe d'une machine virtuelle» à la page 73).

La machine virtuelle ainsi activée portera le nom 'userid' et possédera en standard, les caractéristiques associées par le SEGI à ce userid lorsqu'il a été défini dans le CP (cf. «Définir le hardware de la machine virtuelle» à la page 8).

#### 4.1.2 LOGOFF: arrêter une machine virtuelle

Pour arrêter une machine virtuelle, il suffit d'entrer la commande LOGOFF qui a la syntaxe suivante:

```
Logoff [HOLD]
```

Figure 1. Commande CP LOGOFF

Le paramètre optionnel HOLD permet à un terminal travaillant en commuté<sup>1</sup> de rester en liaison avec le CP après que la machine virtuelle ait été arrêtée.

#### 4.1.3 DISCONN: déconnecter une machine virtuelle

Lorsqu'un travail conséquent ne nécessite pas la présence continue du programmeur devant le terminal, ou encore lorsqu'on désire libérer le terminal pour d'autres utilisateurs sans pour autant arrêter sa machine virtuelle, on peut déconnecter celle-ci à l'aide de la commande suivante:

```
DISConn [HOLD]
```

Figure 2. Commande CP DISCONN

De cette manière, la machine virtuelle demeure présente sans avoir accès au terminal. On peut à tout instant reprendre en main sa machine virtuelle en faisant LOGON sur le terminal d'origine ou sur tout autre terminal. Le paramètre optionnel HOLD remplit la même fonction que pour le LOGOFF. A ce stade deux remarques s'imposent:

1. afin que la machine virtuelle continue à travailler durant la déconnexion, il faut entrer la commande SET RUN ON avant de faire DISC.
2. lorsqu'on reconnecte une machine virtuelle à l'aide de la commande VM1 et si l'on n'avait pas fait SET RUN ON, l'utilisateur se retrouve directement sous le contrôle du CP (ce qui s'intitule communément 'CP console FUNCTION mode'), mais la

<sup>1</sup> Le terme commuté désigne un terminal connecté à un ordinateur du SEGI via une ligne téléphonique.

machine virtuelle est stoppée jusqu'à ce que l'utilisateur entre la commande 'BEGIN'.

#### 4.1.4 Actions automatiques prises par le système

**DISCONNECT automatique** : une machine virtuelle est déconnectée automatiquement par le système d'exploitation lorsque celui-ci détecte une condition d'erreur sur la liaison physique reliant le terminal aux ordinateurs; ce sera notamment le cas si:

1. On effectue un "POWER OFF" du terminal.
2. La connexion est interrompue par une coupure de ligne.
3. Il y a absence de transmission de données pendant une période contiguë de **20 minutes** sur les liaisons qui utilisent le PBX situé au Sart Tilman (dans ce cas, le PBX simule une coupure de ligne. Voir «Concentrateur PBX» à la page 82).

**LOGOFF automatique** : le CP provoque un LOGOFF automatique dans les cas suivants:

1. Lorsqu'une machine déconnectée se trouve dans un état de **CP/VM READ**, situation qui peut se produire lorsque le programme qui s'exécute dans la machine veut effectuer une lecture au terminal ou provoque un ABEND qui bascule la machine virtuelle en état de WAIT; dans ce cas, le LOGOFF est engendré après 15 minutes.
2. Lorsqu'une machine déconnectée se trouve dans un état de **IDLE WAIT STATE**, situation qui peut se produire suite à un WAIT programmé (commande SLEEP par ex.) ou par inactivité de la machine (plus aucun programme ne s'exécute dans la machine); dans ce cas, le LOGOFF est engendré après une période fixée pour les utilisateurs qui auront demandé à bénéficier de cette possibilité.

## 4.2 Définir l'architecture de la machine virtuelle

Par défaut, toutes les machines virtuelles des utilisateurs ont l'architecture XA. Si vous utilisez un programme qui ne supporte pas cette architecture, vous pouvez redéfinir celle-ci au moyen de la commande

```
CP SET MACHINE 370|XA|XC
```

Figure 3. Commande CP SET MACHINE

Après cette commande, il est nécessaire de recharger le système CMS (voir «IPL: charger le système CMS» à la page 13).

## 4.3 Définir le hardware de la machine virtuelle

Ces opérations de définition peuvent se diviser en trois groupes:

1. modifier la taille de la mémoire centrale,
2. définir des minidisques temporaires,
3. établir ou supprimer une connexion vers le minidisque d'une autre machine virtuelle.

En VM le CP possède un fichier appelé DIRECTORY où est décrit le hardware de toutes les machines virtuelles des utilisateurs. Chaque utilisateur possède donc une ou plusieurs machines à sa disposition, identifiées chacune par un nom (userid) qu'il donnera au LOGON afin que le CP sache quelle machine virtuelle démarrer et pour laquelle il trouvera dans la DIRECTORY une description du hardware virtuel associé à cette machine. Tout utilisateur devra donc demander au SEGI de lui définir une machine dans cette DIRECTORY. En standard, le hardware qui lui sera associé a déjà été décrit dans l'introduction. A tout instant on peut obtenir une description complète de la configuration de sa machine virtuelle avec la commande QUERY ALL. Voici donc les commandes CP qui permettront à chacun de modifier ce hardware standard de sa machine après avoir fait LOGON.

### 4.3.1 DEFINE: définir des minidisques temporaires.

Chaque utilisateur peut ajouter à sa machine virtuelle des minidisques temporaires de travail qui viendront en sus de la capacité de stockage disque déjà disponible sur le(s) minidisque(s) de travail donné(s) en standard. Pour obtenir un minidisque temporaire de travail il faut entrer la commande CP suivante:

```
DEFine T3390 vaddr [CYL] nnn
```

Figure 4. Commande CP DEFINE

ce qui a pour effet de connecter un minidisque de type 3390 de nnn cylindres à la machine virtuelle à l'adresse virtuelle vaddr. Chaque machine virtuelle dispose des adresses virtuelles 0000 à 1FFF pour connecter des minidisques temporaires. Cependant il convient de ne pas utiliser les adresses virtuelles déjà occupées et notamment par le hardware virtuel standard prévu dans la DIRECTORY CP c'est-à-dire:

1. la console en 0009
2. le reader en 000C
3. le punch en 000D

4. le printer en 000E
5. le minidisque de travail standard en 0191
6. les minidisques contenant les logiciels publics en 0190, 0192, 019D et 019E.

Un minidisque temporaire est supprimé en faisant LOGOFF ou en faisant DETACH ou DET vaddr où vaddr est l'adresse virtuelle donnée lors du DEFINE.

### 4.3.2 LINK: établir une connexion avec un minidisque d'une autre machine virtuelle

Chaque machine virtuelle possède donc un ou plusieurs minidisques permanents renseignés dans la DIRECTORY du CP où chacun de ces minidisques est associé à une adresse virtuelle donnée. En standard, par exemple, nous avons vu que chaque machine virtuelle possédait un minidisque de travail à l'adresse 191. Chaque machine virtuelle peut établir une connexion vers un minidisque appartenant à une autre machine à l'aide de la commande CP suivante <sup>2</sup> :

```
LINK userid vaddr1 vaddr2 [mode [pw]]
```

Figure 5. Commande CP LINK

Remarque:

- 'userid'** est le nom de la machine propriétaire du minidisque visé,  
**vaddr1** est l'adresse virtuelle du minidisque dans la DIRECTORY de la machine propriétaire  
**vaddr2** est l'adresse virtuelle à laquelle on veut connecter ce minidisque sur sa propre machine. <sup>3</sup>  
**mode** permet de spécifier les modalités d'accès aux minidisques ainsi connectés. Les principales valeurs de **mode** sont les suivantes:
1. **R**: read-only (R/O): le minidisque sera protégé en écriture et la connexion sera refusée si une autre machine a déjà ce minidisque en écriture.
  2. **RR**: read-only (R/O): idem que 'R' mais la connexion est acceptée si une autre machine a déjà ce minidisque en écriture.
  3. **WR**: read-write (R/W): la connexion est établie en lecture-écriture mais si une autre machine accède déjà ce minidisque (en lecture ou en écriture), la connexion sera établie en lecture seulement.
  4. **M**: multiple access (R/W): la connexion est établie en lecture-écriture sauf si une autre machine accède déjà au minidisque en écriture. Dans ce cas la connexion est refusée.

**pw** permet de spécifier le password associé au minidisque dans la DIRECTORY pour le mode d'accès indiqué.

Si la connexion est établie en lecture seulement, il faut donner le read-password qui est d'ailleurs optionnel si le password est 'ALL'. Si la connexion est établie en écriture, il faut donner le write-password. Cette technique des passwords permet donc à chaque propriétaire de machine virtuelle de limiter les accès d'autrui à ses minidisques.

En standard, dès que l'on fait LOGON, CP établit automatiquement un LINK

- vers chacun des minidisques dont la machine est propriétaire; ces minidisques sont connectés aux adresses définies dans la DIRECTORY (par exemple le 191);
- vers 3 minidisques qui contiennent des logiciels publics au moyen des commandes suivantes :

```
LINK MAINT 190 190 RR
LINK MAINT 19D 19D RR
LINK MAINT 19E 19E RR
LINK PA 319 192 RR
```

Une connexion est supprimée par le LOGOFF ou par une commande CP DETACH ou DET vaddr où vaddr est l'adresse virtuelle vaddr2 donnée au LINK.

Remarques:

1. le password n'est jamais requis pour établir une connexion vers l'un de ses propres minidisques;
2. si le mode n'est pas spécifié, on aura par défaut 'R' si on demande une connexion vers un minidisque d'une autre machine virtuelle, 'M' si on demande une connexion vers l'un de ses propres minidisques;
3. la meilleure protection consiste à n'introduire aucun password, ce qui interdit toute connexion par la commande link.

## 4.4 Gestion des unit records

Nous avons vu que le punch et le printer d'une machine virtuelle envoient les decks et les listings dans le SPOOL. Le reader d'une machine virtuelle permet lui d'extraire des cartes virtuelles (de 80 à 132 colonnes) du SPOOL et de les mettre sur un device quelconque (disque, terminal, printer, punch).

Le SPOOL est logiquement subdivisé selon différents critères dont voici le détail:

1. d'après le propriétaire du fichier (userid)

<sup>2</sup> Pour autant qu'elle y soit autorisée par le système de sécurité. Voir «Gestion et protection des minidisques» à la page 74.

<sup>3</sup> On peut également faire un LINK vers l'un de ses propres minidisques afin de l'utiliser à une autre adresse virtuelle que celle définie dans la DIRECTORY.

- d'après le type de UR (reader, punch, printer)
- d'après la classe (A à Z et 0 à 9)<sup>4</sup>
- d'après le créateur (originid=userid de celui qui a créé le fichier).

Pour plus de simplicité nous considérons que chaque paquet de cartes, chaque listing situé dans le SPOOL constitue un 'fichier spoolé' auquel est associé un userid et une classe, et qu'un fichier est dans le reader, sur le punch ou sur l'imprimante de la machine virtuelle 'userid'.

En outre, on peut associer à chaque fichier spoolé un nom et un type de huit caractères chacun, ainsi qu'un code de destination (distribution code) de huit caractères également qui apparaîtra sur tout listing ou deck réels afin notamment d'en simplifier la restitution par les opérateurs. Enfin, chaque fichier spoolé se voit affecté par le CP un 'spoolid' qui est un numéro de quatre chiffres: chaque fichier spoolé reçoit ainsi un numéro unique permettant de le distinguer de tout autre.

Pour connaître les caractéristiques de ses UR, il suffit d'entrer la commande CP:

```
Query UR ALL
```

**Figure 6. Commande CP QUERY**

tandis que pour connaître le détail des fichiers situés sur son reader, punch ou printer, il faut entrer respectivement les commandes CP suivantes:

```
Query Reader ALL
Query PUnch ALL
Query Printer ALL
```

**Figure 7. Commande CP QUERY**

Par défaut, les fichiers (en classe A ou T) envoyés sur l'imprimante de la machine virtuelle sont repris en main par le CP qui les imprime sur une imprimante réelle.

Nous venons de décrire le contenu du SPOOL, il nous reste maintenant à voir tout ce que l'on peut en faire et vous verrez que le SPOOL est une véritable plaque tournante qui permet également aux machines virtuelles d'échanger des informations.

En effet, le SPOOL permet d'effectuer les transferts suivants:

- transférer un fichier d'un printer vers un reader (maximum 132 caractères par enregistrement),

- transférer un fichier d'un punch vers un reader (maximum 80 caractères par enregistrement).

#### 4.4.1.1 SPOOL: destination d'un fichier printer ou punch

Afin de destiner à un reader un fichier que l'on va imprimer ou perforer, il faut, avant d'envoyer des informations sur le printer ou le punch, entrer l'une des commandes cp suivantes:

```
SPOOL { Printer
        PUnch } [userid
                *
                SYSTEM] [CONT
                       NOCONT] [CLass c] [HOLD
                                           NOHOLD]
```

**Figure 8. Commande CP SPOOL**

Ce qui a pour effet d'associer au périphérique pour tout fichier ultérieurement envoyé sur le printer (ou le punch) une information destinée au CP disant que ce fichier doit être mis sur le reader de la machine 'userid' en classe 'c' (si la classe n'est pas spécifiée, elle reste ce qu'elle était avant la commande SPOOL), soit sur le reader de sa propre machine (option \*) en classe 'c', soit envoyé sur l'imprimante réelle du CP (option 'SYSTEM') en classe 'c' pour être imprimé. L'option 'CONT' signifie que tous les fichiers envoyés sur le printer ou le punch seront regroupés en un seul fichier dans le SPOOL et cela jusqu'à ce que l'on fasse LOGOFF ou une commande CP:

```
CLose { Printer
        PUnch } [NAme fn] EOF
```

ou

```
SPOOL { Printer
        PUnch } CLOSE
```

**Figure 9. Commandes CP CLOSE et SPOOL**

L'option HOLD place en hold status tous les fichiers créés ultérieurement sur le device. Cet état de hold peut être modifié avec la commande CHANGE. Les fichiers envoyés en hold status sur un reader ne pourront être lus qu'après avoir retiré ce hold status. Les fichiers envoyés sur le printer SYSTEM ne sont pas imprimés s'ils ont le hold status: il faudra donc le retirer pour le(s) laisser partir en impression.

<sup>4</sup> Le fait de pouvoir associer différentes classes aux fichiers situés dans le spool permet de faire des regroupements logiques: ainsi nous verrons que dans certaines commandes on peut indiquer une classe afin de ne travailler qu'avec les fichiers de la classe indiquée. En outre, pour imprimer un fichier on doit nécessairement le mettre en classe A ou T.

## REMARQUES:

1. Les fichiers destinés à être imprimés doivent obligatoirement être en classe A ou T (cfr. «Conservation des fichiers dans un spool VM» à la page 86)

2. Certains paramètres de la commande **SPOOL Printer** permettent de modifier les caractéristiques d'impression affectées, par défaut, aux fichiers. C'est ainsi que:

**COpy nnn** permet de demander l'impression de nnn copies d'un même fichier.

3. On peut éviter de voir disparaître systématiquement du SPOOL tout fichier situé sur un reader virtuel dès qu'il a été lu en faisant, AVANT l'opération de lecture:

```
SPOOL Reader HOLD
```

**Figure 10. Commande CP SPOOL READER HOLD**

On retire l'état de hold associé ainsi au reader en faisant

```
SPOOL Reader NOHOLD
```

**Figure 11. Commande CP SPOOL READER NOHOLD**

4. Pour extraire un fichier du reader, il faut que ce dernier soit spoolé dans la classe de ce fichier. Pour affecter une classe au reader on utilise la commande

```
SPOOL Reader CLass {x*}
```

**Figure 12. Commande CP SPOOL READER CLASS**

où x indique la classe choisie et \* indique la sélection de toutes les classes.

Au logon, le CP exécute automatiquement les commandes:

```
SPOOL PRINTER SYSTEM NOCONT CLASS A  
SPOOL PUNCH SYSTEM NOCONT CLASS A
```

Donc si l'utilisateur de la machine USER2 veut envoyer ses trois fichiers F1, F2 et F3 dans le reader de la machine USER1 en utilisant le punch, il fera:

```
SPOOL PU USER1 CONT
```

```
punch de F1, puis de F2, puis de F3  
SPOOL PU NOCONT  
CLOSE PU NAME F1F2F3 FUSER2
```

PUNch est une commande CMS qui sera présentée ultérieurement («Envoyer un fichier sur le punch» à la page 20).

Nous verrons par la suite qu'il existe une méthode beaucoup plus simple pour échanger des fichiers entre machines virtuelles (cfr. «Transmission de fichiers.» à la page 37).

Après avoir effectué ces opérations, USER1 recevra sur son reader un fichier en classe A qui est la concaténation des fichiers F1, F2 et F3. Lorsque USER1 fera la commande CP : Q R ALL il recevra un message lui donnant:

1. l'origine du fichier spoolé (USER2)
2. le spoolid de ce fichier
3. la classe (A)
4. l'UR utilisé pour le transfert (PUNCH)
5. le nombre de records
6. le nombre de copies
7. le "hold status"
8. la date et l'heure de la commande CP CLOSE
9. le nom et le type du fichier (F1F2F3 FUSER2)
10. le distribution code de USER2

Il lui suffira donc alors de le lire pour le mettre sur l'un de ses minidisques.

### 4.4.1.2 PURGE: éjecter des fichiers du SPOOL

La commande CP PURGE permet d'éjecter du spool les fichiers dont on n'a plus l'usage.

```
PURge qname id
```

**Figure 13. Commande CP PURGE**

où qname indique la queue sur laquelle est situé le fichier (Printer, PUNch ou Reader) et id identifie le fichier visé; celui-ci peut être défini soit:

<b>par la classe</b>	Class c1
<b>par le spoolid</b>	spoolid
<b>sans critère</b>	ALL

1. Pour vider complètement son reader on fera donc PUR R ALL
2. Pour éjecter du punch les fichiers de classe A on fera PUR PU CL A

Remarque: les fichiers éjectés du spool sont irrémédiablement perdus.

#### 4.4.1.3 CHANGE: changer les attributs d'un fichier spoolé

Lorsqu'un fichier a été spoolé sur le punch, le printer ou le reader, on peut modifier sa classe, son distribution code, son nom, son type et son hold status au moyen de la commande CP suivante:

```
CHange qname id [Class c2][NA fn ft]  $\left[ \begin{array}{l} \text{HOLD} \\ \text{NOH} \end{array} \right]$  [DI dist]
```

Figure 14. Commande CP CHANGE

où qname indique la queue sur laquelle est situé le fichier cad Printer, PUnch ou Reader et id identifie le fichier visé soit

par la classe	Class c1
par le spoolid	spoolid
sans critère	ALL

Pour changer les fichiers spoolés sur le punch en classe A et les mettre en classe B, on fera donc: CH PU CL A CL B

#### 4.4.1.4 TAG: préciser la destination finale d'un fichier spoolé

En VM, il existe une machine déconnectée appelée RSCS (voir «Réseau networking» à la page 78) en permanence active, qui permet d'envoyer des fichiers vers n'importe quelle imprimante ou ordinateur connectés au système VM du SEGI.

Pour envoyer un fichier vers l'une de ces destinations il faut l'envoyer dans le reader de RSCS mais au préalable il faut lui associer, au moyen de la commande CP TAG, une information précisant la destination.

La commande CP TAG a la syntaxe suivante:

```
TAg Dev qname tagtext
```

Figure 15. Commande CP TAG

Remarque:

<b>qname</b>	indique la queue à laquelle doit être associé le tagtext
<b>'tagtext'</b>	est l'information de destination à associer aux fichiers envoyés sur le printer ou le punch (voir «Réseau networking» à la page 78).

#### 4.4.1.5 TRANSFER: récupérer un fichier spoolé vers une autre machine

Nous avons vu que les commandes CHANGE, PURGE et SPOOL permettent de gérer le spool. Mais, une machine virtuelle ne peut modifier les caractéristiques d'un fichier spoolé que si ce fichier se trouve sur un de ses units records (reader, printer ou punch), c'est-à-dire si elle en est le *propriétaire*.

Lorsque la machine USER1 crée et envoie un fichier sur le reader de la machine USER2, c'est USER2 qui en devient le propriétaire et c'est donc lui seul qui peut en modifier les caractéristiques. Par contre, USER1 demeure le *créateur* du fichier et à ce titre il peut demander au CP de lui en rendre la propriété au moyen de la commande TRANSFER (pour autant bien entendu qu'entretemps USER2 n'ait pas fait disparaître le fichier : PURGE,...). Chaque fichier spoolé a donc un et un seul créateur ainsi qu'un et un seul propriétaire à un moment donné.

```
TRANsfer qname id TO userid  $\left[ \begin{array}{l} \text{Printer} \\ \text{Reader} \end{array} \right]$ 
```

Figure 16. Commande CP TRANSFER qname

Remarque: qname indique la queue sur laquelle est situé le fichier cad Printer, PUnch ou Reader et id identifie le fichier visé soit

par la classe	Class c1
par le spoolid	spoolid
sans critère	ALL

L'option TO est utilisée lorsque le requérant veut transmettre son titre de propriété à autrui cad transférer un fichier de son Unit Record vers le Unit Record d'une autre machine.

```
TRANsfer Reader id FROM  $\left\{ \begin{array}{c} * \\ \text{userid} \\ \text{ALL} \end{array} \right\}$  qname
```

Figure 17. Commande CP TRANSFER READER

L'option FROM est utilisée lorsque le créateur d'un fichier spoolé vers autrui veut récupérer ce dernier.

Le changement de propriété peut être assorti d'un changement de 'queue': ainsi, tout fichier spoolé sur un punch ou un printer peut être transféré sur un reader et tout fichier spoolé sur un reader peut être transféré sur sa 'queue' d'origine (punch ou printer).

Les options spoolid et CLASS permettent de limiter l'opération au(x) seul(s) fichier(s) portant le spoolid donné ou spoolés dans la classe donnée.

Exemple:

Imaginons que USER1 ait exécuté en interactif un programme qui envoie les résultats sur son printer virtuel en hold et en classe T. Avant de laisser imprimer le listing, il décide de vérifier ces résultats: il va donc transférer ce fichier du printer vers son reader en faisant:

```
TRAN R CL T FROM USER1 P
```

puis il met le reader en hold status et retire le hold status associé au fichier afin de pouvoir le lire sans qu'il disparaisse du reader:

```
SP R HOLD  
CH R CL T NOHOLD
```

il peut alors le lire puis, s'il le désire, le faire imprimer en faisant:

```
TRAN CL T TO * P
```

Remarque: si USER1 avait plusieurs fichiers spoolés en classe T, il aurait intérêt à utiliser l'option spoolid plutôt que l'option CLASS afin de ne manipuler que le fichier visé plutôt que tous les fichiers de classe T.

## 4.5 IPL: charger le système CMS

Lorsqu'on dispose d'une machine virtuelle, encore faut-il la charger avec le système CMS avant de pouvoir exécuter des programmes. Nous avons vu dans l'introduction pourquoi le SEGI a choisi le système CMS plutôt qu'un operating system comme MVS sous VM. Pour charger CMS on entre la commande CP

```
Ipl CMS
```

Figure 18. Commande CP IPL

Après avoir fait cela votre machine virtuelle est à même d'exécuter tous les travaux prévus pour tourner sous CMS.

Dans une machine virtuelle, on peut, à tout instant, réinitialiser le système CMS en refaisant IPL CMS. C'est un moyen radical pour nettoyer toutes traces d'un travail antérieur.

Remarque : en standard, au logon, CP fait automatiquement une commande IPL CMS.

## 4.6 Questionner le CP sur l'état de la machine virtuelle.

A tout instant on peut questionner le CP pour obtenir des renseignements sur l'état de notre machine virtuelle à l'aide de la commande:

```
INDicate USER
```

Figure 19. Commande CP INDICATE USER

On obtient notamment les indications suivantes :

- userid** userid de la machine virtuelle.
- MACH** architecture de la machine virtuelle, 370 ou XA.
- STOR** taille de la mémoire virtuelle en Kbytes ou Mbytes.
- RES** le nombre de blocs de 4k de mémoire centrale réelle occupés par votre machine.
- CTIME** temps de connexion depuis le LOGON exprimé en heures et minutes.
- VTIME** CPU virtuel consommé depuis le LOGON exprimé en minutes et secondes. Le VTIME comprend le CPU scalaire aussi bien que vectoriel.
- TTIME** CPU total (c'est-à-dire VTIME augmenté de l'"overhead" du CP) consommé depuis le LOGON exprimé en minutes et secondes.
- IO** le nombre d'opérations d'entrées/sorties (à l'exclusion des accès au SPOOL) depuis le début de la période de facturation en cours (changement de période à 7h et 18h).
- CTIMEA, VTIMEA, TTMEA** ont le même sens que CTIME, VTIME et TTIME, mais donnent les consommations depuis le début de la période de facturation en cours.
- RDR** le nombre de cartes virtuelles lues depuis le début de la période de facturation en cours.
- PRT** le nombre de lignes virtuelles écrites depuis le début de la période de facturation en cours.
- PCH** le nombre de cartes virtuelles perforées depuis le début de la période de facturation en cours.
- IOT** le nombre d'opérations d'entrées/sorties (à l'exclusion des accès au SPOOL) depuis le LOGON.
- VVECTIME** CPU virtuel vectoriel consommé depuis le LOGON exprimé en minutes et secondes.
- TVECTIME** CPU total vectoriel consommé depuis le LOGON.
- VVECTIMEA, TVECTIMEA** ont le même sens que VVECTIME et TVECTIME mais donnent les consommations depuis le début de la période de facturation en cours (changement de période à 7h et 18h).
- ...

Ou encore

```

Query Time
Query TERMinal
Query DAsd
Query STORage
Query Links nnn
Query Reader
Query Printer
Query PUnch
Query PF
Query Set
Query ALL
...

```

**Figure 20. Commande CP QUERY**

**Time** l'heure, le jour, la date, ainsi que le temps de connexion et de CPU utilisés par votre machine depuis le LOGON.

**TERMinal** les options en effet pour votre console virtuelle.

**DAsd** la liste et le mode d'accès des minidisques connectés.

**STORage** la taille de mémoire virtuelle de votre machine.

**Links** le USERID et le mode d'accès de toutes les machines qui sont connectées au minidisque dont l'adresse virtuelle est donnée par nnn.

**Reader** une description de tous les fichiers situés sur votre lecteur virtuel.

**Printer** une description de tous les fichiers situés sur votre imprimante virtuelle (ils disparaissent dès qu'ils sont en impression).

**PUnch** une description de tous les fichiers situés sur votre perforateur de cartes virtuel.

**PF** une description des commandes associées aux touches fonctions.(cfr«Utilisation des touches fonctions.»)

## 4.7 Facilités particulières du CP

### 4.7.1 Envoyer des messages.

La commande CP:

```
MSG { userid } msgtxt
      { OPERator }
```

**Figure 21. Commande CP MSG**

permet d'envoyer le message 'msgtxt' à la machine 'userid' ou à l'opérateur (OPERATOR).

Il existe d'autres manières d'échanger des messages (cfr. « Echange de messages entre machines virtuelles: TELL» à la page 36).

### 4.7.2 Utilisation des touches fonctions.

Le CP permet également d'associer une quelconque commande à une touche PFK au moyen de la commande (cfr. «Rappel de commandes.» à la page 15).

```
SET PFnn [IMMed] pfdata
```

**Figure 22. Commande CP SET**

**PFnn** est la touche PFK à laquelle on veut associer la commande pfdata.

**'IMMED'** signifie que, dès que l'on utilise la PFK, la commande associée sera exécutée. Si 'IMMED' est omis, la commande associée sera listée à l'écran, ce qui laisse à l'utilisateur la possibilité de la modifier avant de pousser sur la touche 'ENTER' pour l'exécuter. Cette commande est valable jusqu'au prochain SET ou jusqu'au LOGOFF.

Si l'on fait:

```
SET PF1 IMMED Q P ALL
```

chaque fois que l'on poussera sur la touche PF1, la commande Q P ALL sera exécutée.

### 4.7.3 Messages d'erreur.

La commande CP SET EMSG ON permet d'obtenir pour chaque message d'erreur produit, le numéro sous lequel ce message est repris dans la documentation et dans le HELP (Voir « La commande HELP.» à la page 43).

### 4.7.4 Définition du terminal.

La commande TERMINAL a la forme suivante:

```
TERminal opt1 opt2
```

**Figure 23. Commande CP TERMINAL**

**opt1** CHardel ou EEscape ou LINEDel ou LINENd ou TABchar ou APL ou LINEsize

**opt2** ON ou Off ou nnn

Elle permet de contrôler les fonctions suivantes associées au terminal; possibilités pouvant être rendues inopérantes (option OFF) et ensuite opérationnelles (option ON) :

- définir un caractère (CH char) de suppression logique du caractère précédent entré au clavier;
- définir un caractère (ES char) de validation logique (escape) du caractère suivant entré au clavier;<sup>5</sup>
- définir un caractère (LINEDEL char) de suppression logique de la ligne entrée au clavier;
- définir un caractère séparateur de lignes (LINEND char);
- définir un caractère (TAB char) de tabulation logique.

Autres possibilités

- définir la longueur des lignes affichées à l'écran : LINES nnn (1<nnn<255);
- APL (voir «Bibliographie» à la page 113).

Sur un terminal 3270, dès le logon, la commande TERMINAL suivante est exécutée automatiquement:

```
TERM CH OFF ES OFF LINEN # LINED OFF TAB OFF LINES 80
```

**Figure 24. Commande CP TERM CH**

Certains packages requièrent de faire une commande TERMINAL particulière: la commande requise est notée dans le chapitre relatif à chacun de ces produits.

### 4.7.5 Rappel de commandes.

Sur les terminaux 3270 ou assimilés il est possible de définir une touche fonction qui permettra de rappeler à l'écran les 15 dernières commandes envoyées afin d'éviter de les réencoder pour les exécuter à nouveau.

```
SET PFnn RETRIEVE [ BACKWARD ]  
                   [ FORWARD ]
```

**Figure 25. Commande CP SET PFnn**

On retrouve les commandes dans l'ordre de la plus récente vers la plus ancienne avec BACKWARD et dans l'ordre inverse avec FORWARD.

---

<sup>5</sup> ainsi si vous désirez entrer dans une commande un caractère défini comme CHardel vous devrez le précéder du caractère d'EScape afin qu'il soit considéré comme donnée et non comme CHardel.



## 5.1 Utilisation d'un minidisque en CMS

Les minidisques sont connectés à une machine virtuelle via une adresse virtuelle. Pour accéder à un minidisque sous CMS, il faut également lui associer un 'filemode letter' qui est une des lettres A à Z.

Nous avons vu qu'après le logon, CP établit automatiquement une connexion vers 4 minidisques situés aux adresses virtuelles 190, 191, 192, et 19E. Dès que la commande IPL CMS est exécutée, CMS va affecter automatiquement un 'filemode letter' à chacun de ces minidisques comme suit:

minidisque	fm letter
0191	A
0192	D
0190	S
019E	Y

Pour assigner un fm letter à un minidisque connecté il faut utiliser la commande CMS

```
ACcess vaddr fm-letter
```

**Figure 26. Commande CMS ACCESS**

Exemple: AC 193 E

En CMS, on parlera de disque A lorsqu'on envisage le minidisque associé au fm-letter A.

Dans les cas où CMS parcourt les minidisques pour trouver un fichier, il effectue sa recherche dans l'ordre alphabétique des fm-letter.

Avant de pouvoir mettre des informations sur un minidisque connecté, il faut au préalable le formater à l'aide de la commande CMS

```
FORMAT vaddr fm-letter
```

**Figure 27. Commande CMS FORMAT**

En général, on ne formate un minidisque que la toute première fois que l'on désire l'utiliser (la première fois que vous ferez LOGON, vous devrez donc formater votre disque A).

Remarque: la commande FORMAT vaddr fm-letter engendre automatiquement une commande AC vaddr fm-letter.

Sous CMS, l'on peut demander la liste des minidisques accédés à un moment donné en faisant la commande CMS:

```
Query DISK
```

**Figure 28. Commande CMS QUERY DISK**

L'inverse de la commande CMS ACCESS est la commande:

```
RELease fm-letter [ (DET )
```

**Figure 29. Commande CMS RELEASE**

Si l'on spécifie 'DET', l'access est supprimé et en plus la connexion vers le minidisque est retirée (Voir «Définir le hardware de la machine virtuelle» à la page 8).

**Remarque :** Il existe deux commandes qui permettent de définir et formater des minidiques temporaires : TDSKCMS et TDSKOS.

La définition d'un minidisque CMS peut se faire via la commande suivante:

```
TDSKCMS cnn nbcyl [fm]
```

**Figure 30. Commande EXEC TDSKCMS**

où

**cnn** adresse du minidisque CMS à définir  
**nbcyl** nb de cylindres du minidisque  
**fm** filemode avec lequel ce minidisque sera formaté et donc accédé: Par défaut, fm = le 1er filemode disponible

Cette commande remplace les deux suivantes :

```
CP DEFINE T3390 ...
FORMAT ...
```

TDSKOS, uniquement disponible sur le superordinateur, permet de définir et formater un minidisque VSAM :

```
TDSKOS cuu nbcyl [valid]
```

Figure 31. Commande EXEC TDSKOS

où

**cuu** adresse du minidisque OS à définir  
**nbcyl** nombre de cylindres du minidisque  
**valid** nom du volume (optionnel)

Remarques:

- Le minidisque ainsi défini sera initialisé OS avec une VTOC en cylindre 0.
- Cette commande peut être utilisée aussi sous cmsbatch.

### 5.1.1 Nomenclature des fichiers CMS

En CMS, les fichiers sont identifiés par un filename (fn) et un filetype (ft) de 8 caractères alphanumériques chacuns (A-Z, 0-9 et \$, #, @) et un filemode (fm) de 2 caractères: le premier est une lettre de A à Z, le second est un chiffre de 0 à 6 (le filemode number).

En général, chaque utilisateur choisit le fn qu'il désire, mais par contre, le ft est souvent imposé :

<b>ft ASSEMBLE</b>	identifie un fichier contenant un programme source assembler,
<b>ft FORTRAN</b>	identifie un fichier contenant un programme source fortran,
<b>ft PLIOPT et PLI</b>	identifient un fichier contenant un programme source PL/I,
<b>ft TEXT</b>	identifie un fichier contenant un module objet,
<b>ft LISTING</b>	identifie un fichier d'impression ayant un code de contrôle d'avancement de papier en tête de chaque enregistrement,
<b>ft MODULE</b>	identifie un fichier contenant un load module CMS,
<b>ft DATA</b>	identifie un fichier contenant les données d'un programme d'application,
<b>ft EXEC</b>	identifie un fichier contenant une procédure de commandes,
<b>ft MACLIB</b>	identifie un fichier contenant une bibliothèque de macros,
<b>ft TXTLIB</b>	identifie un fichier contenant une bibliothèque de modules objets,

<b>ft LOADLIB</b>	identifie un fichier contenant une bibliothèque de modules OS.
<b>ft SCRIPT</b>	identifie un fichier contenant du texte en format DCF:
<b>ft LIST3820</b>	identifie un fichier d'impression laser.
<b>ft PSEG38PP</b>	identifie un fichier graphique laser bit map.
<b>ft GDF</b>	identifie un fichier graphique GDDM vectorisé.
<b>ft COBOL</b>	identifie un fichier source cobol.

Le filemode letter identifie le minidisque sur lequel se trouve le fichier: le filemode letter A signifie que le fichier est situé sur le disque A.

Le filemode number est utilisé comme suit:

<b>0</b>	les machines virtuelles qui ne sont pas propriétaires du minidisque ne peuvent avoir accès au fichier que si elles ont accès au minidisque en lecture/écriture.
<b>1</b>	c'est le filemode number habituellement utilisé et pris par défaut chaque fois que l'on crée un fichier.
<b>2 et 5</b>	ont la même utilité que 1. Ils permettent simplement à chacun d'établir une sous-distinction dans la répartition de ses fichiers sur un même minidisque.
<b>3</b>	les fichiers avec le fm number 3 sont automatiquement effacés après avoir été relus.
<b>4</b>	réserve pour les 'OS simulated datasets': ce sont des fichiers qui ne sont généralement pas créés par des commandes CMS mais par des programmes qui utilisent des macro-OS (cfr «Programmation sous CMS» à la page 45).
<b>6</b>	est utile, en particulier, pour définir un minidisque qui sera réservé à un seul fichier. Les blocs du minidisque correspondent aux enregistrements du fichiers et sont numérotés à partir de 1. Ils sont ainsi accessibles en accès direct. Après que le minidisque a été formaté, la définition du fichier se fait au moyen de la commande suivante :

```
RESERVE fn ft fm
```

Figure 32. Commande CMS RESERVE

Exemple: le fichier 'TEST FORTRAN A1'

- réside sur le disque A (fm letter = A)
- possède le fm number 1
- contient un programme source fortran (ft = FORTRAN)
- s'appelle TEST (fn = TEST).

## 5.1.2 Ordre de balayage des minidisques

Lorsque CMS parcourt les minidisques accessibles à la recherche d'un fichier, il le fait dans l'ordre alphabétique des filemode letters: CMS cherche donc d'abord sur le disque A, puis sur le disque B, puis sur le disque C, etc...

## 5.1.3 Commandes CMS permettant d'accéder aux fichiers sur minidisques

### 5.1.3.1 LISTFILE

$$\text{Listfile} \left[ \begin{array}{c} \text{fn} \\ * \\ - \end{array} \left[ \begin{array}{c} \text{ft} \\ * \\ - \end{array} \left[ \begin{array}{c} \text{fm} \\ \text{A} \\ * \end{array} \right] \right] \right] \text{[(L)]}$$

Figure 33. Commande CMS LISTFILE

Remarque: le fm A est pris par défaut

Cette commande permet d'avoir la liste des fichiers accessibles répondant aux critères fn ft fm. Si l'option '(L' a été spécifiée, on a en plus pour chacun d'eux, le format des enregistrements, la longueur maximale des enregistrements, le nombre d'enregistrements, la date et l'heure de création.

\* signifie qu'il n'y a pas de critère de sélection.

Exemples :

**L** est équivalent à :

L \*

ou à L \* \*

ou à L \* \* A

et donne la liste de tous les fichiers situés sur le disque A.

**L TEST** est équivalent à :

L TEST \*

ou à L TEST \* A

et donne la liste de tous les fichiers situés sur le disque A et dont le fn est TEST.

**L T\*** est équivalent à :

L T\* \*

ou à L T\* \* A

et donne la liste de tous les fichiers situés sur le disque A et dont le fn commence par T.

**L \* TEXT \*** donne la liste de tous les fichiers accessibles (c'est-à-dire situés sur n'importe quel disque accessible) et dont le ft est TEXT.

### 5.1.3.2 Lister un fichier :

$$\text{Type fn ft} \left[ \begin{array}{c} \text{fm} \\ \text{A} \\ * \end{array} \right] \text{[(MEM \{ name \} *)]}$$

Figure 34. Commande CMS TYPE

Remarque: le fm A est pris par défaut. Cette commande permet de lister le fichier fn ft fm au terminal.

'(MEM \*' permet de lister tous les membres d'une bibliothèque<sup>6</sup> '(MEM name' permet de ne lister que le member 'name' de la bibliothèque.

Exemples :

- T TEST FORTRAN est équivalent à T TEST FORTRAN A et permet de lister le fichier TEST FORTRAN situé sur le disque A.
- T TEST FORTRAN \* permet de lister le premier fichier TEST FORTRAN rencontré (cfr ordre de balayage des minidisques).

### 5.1.3.3 Supprimer un fichier :

$$\text{ERASE} \left[ \begin{array}{c} \text{fn} \\ * \end{array} \right] \left[ \begin{array}{c} \text{ft} \\ * \end{array} \right] \left[ \begin{array}{c} \text{fm} \\ \text{a} \\ * \end{array} \right] \text{[( Type)]}$$

Figure 35. Commande CMS ERASE

permet de supprimer tous les fichiers correspondants aux critères fn ft fm. L'option '(T' permet d'obtenir la liste des fichiers supprimés).

<sup>6</sup> La notion de bibliothèque est décrite dans «Gérer les bibliothèques MACLIB, TXTLIB et LOADLIB» à la page 22

### 5.1.3.4 Changer le nom d'un fichier :

$$\text{Rename } \left\{ \begin{matrix} \text{fn1} \\ * \end{matrix} \right\} \left\{ \begin{matrix} \text{ft1} \\ * \end{matrix} \right\} \left\{ \begin{matrix} \text{fm1} \\ * \end{matrix} \right\} \left\{ \begin{matrix} \text{fn2} \\ = \end{matrix} \right\} \left\{ \begin{matrix} \text{ft2} \\ = \end{matrix} \right\} \left\{ \begin{matrix} \text{fm2} \\ = \end{matrix} \right\} [(\text{Type})]$$

Figure 36. Commande CMS RENAME

Remarque: le fm A est pris par défaut. Cette commande permet de changer le nom des fichiers correspondants aux critères fn1 ft1 fm1 en fn2 ft2 fm2. L'option '(T' permet d'obtenir la liste des fichiers changés. Le symbole '=' indique que la zone correspondante ne doit pas être changée.

Exemple:

```
R * DATA1 A1 = DATA2 A2
```

pour changer le ft et le fm de tous les fichiers situés sur le disque A avec le ft DATA1 et le fm number 1 en DATA2 A2.

TEST DATA1 A1 deviendra donc TEST DATA2 A2.

### 5.1.3.5 Copie fichier à fichier :

$$\text{COPYfile fn1 ft1 fm1 fn2 ft2 fm2} \left[ \begin{matrix} \text{REPlace} \\ \text{APpend} \end{matrix} \left[ \text{OLDD} \right] \right]$$

Figure 37. Commande CMS COPYFILE

permet de copier le fichier fn1 ft1 fm1 dans le fichier fn2 ft2 fm2. Si fn2 ft2 fm2 existait déjà il faut spécifier, soit l'option '(REP' pour remplacer le contenu de fn2 ft2 fm2 par le contenu de fn1 ft1 fm1, soit l'option '(AP' pour ajouter le contenu de fn1 ft1 fm1 au fichier fn2 ft2 fm2. Le paramètre OLDD permet de conserver la date et l'heure de création/mise à jour du fichier original.

### 5.1.3.6 Envoyer un fichier sur le printer

$$\text{PRint fn ft} \left[ \begin{matrix} \text{fm} \\ \text{A} \\ * \end{matrix} \right] \left[ \left( \left[ \text{MEM} \left\{ \begin{matrix} \text{name} \\ * \end{matrix} \right\} \right] \right) \left[ \text{CC} \right] \left[ \text{UP} \right] \right]$$

Figure 38. Commande CMS PRINT

Remarque: le fm A est pris par défaut.

Cette commande permet d'envoyer le fichier fn ft fm sur le printer. Si fm vaut \* la séquence normale de balayage est effectuée. L'option '(MEM name' permet d'envoyer le membre 'name' d'une bibliothèque MACLIB ou TXTLIB (voir «Gérer les bibliothèques MACLIB, TXTLIB et LOADLIB» à la page 22). L'option '(MEM \*' permet d'envoyer tous les membres de la bibliothèque.

L'option CC permet de signaler à la commande PRINT que le fichier contient des caractères de contrôle d'avancement de papier en colonne 1 (CC est pris par défaut lorsque le ft du fichier est LISTING ou SCRIPT).

L'option UP permet de convertir les minuscules non accentuées en majuscules.

### 5.1.3.7 Envoyer un fichier sur le punch

$$\text{PU} \text{ nch fn ft} \left[ \begin{matrix} \text{fm} \\ \text{A} \\ * \end{matrix} \right] \left[ \left( \left[ \text{MEM} \left\{ \begin{matrix} \text{name} \\ * \end{matrix} \right\} \right] \right) \left[ \text{NOHeader} \right] \right]$$

Figure 39. Commande CMS PUNCH

Remarque: le fm A est pris par défaut.

Cette commande est identique à PRINT sauf pour l'option NOHEADER. Si l'option NOHEADER est omise, le fichier envoyé sur le punch sera précédé d'une carte d'identification où l'on retrouve le fn, le ft et le fm du fichier d'origine. Cette carte d'identification appelée HEADER est utilisée par certaines commandes CMS qui lisent un fichier sur le reader afin de retrouver le fn, ft et fm original.

### 5.1.3.8 Lire un fichier situé sur le reader: PEEK

La commande PEEK permet d'afficher un fichier situé sur le reader virtuel sans devoir le lire sur un minidisque. La commande PEEK vous place en mode XEDIT (Voir «L'édition en CMS» à la page 25) ce qui permet d'utiliser toutes les facilités de ce dernier pour le parcourir ou le sauver sur disque.

le format de la commande PEEK est le suivant:

$$\text{PEEK [spoolid] } \left[ \left( \left[ \text{FRom recno} \right] \left[ \text{FOr numrec} \right] \right) \right]$$

Figure 40. Commande EXEC PEEK

Remarque:

**spoolid** est le numéro du fichier à afficher. Par défaut ce sera le premier fichier non holdé du reader dont la classe a été reprise dans la dernière commande SPOOL READER CLASS x.

**recno** indique à partir de quelle ligne du fichier on doit effectuer l'affichage. Par défaut ce sera à partir du début du fichier.

**numrec** représente le nombre de lignes à afficher. Par défaut ce sera 200. Si numrec vaut \* tout le fichier sera affiché.

Quoi que vous fassiez du fichier édité avec les commandes XEDIT, son état sur le reader ne sera pas altéré.

On peut effacer le fichier du reader en entrant la commande DISCARD sous XEDIT

Dès que le fichier est affiché les touches fonctions reçoivent une affectation décrite au bas de l'écran et explicitée dans le chapitre relatif à XEDIT. (Voir «L'édition en CMS» à la page 25)

**Note** : Lors de l'examen par PEEK d'un fichier envoyé dans le reader par la commande SENDFILE, le formatage ne sera pas correct si une partie seulement du fichier est lue (par défaut, les 200 premières lignes). Pour avoir un formatage correct, on peut :

soit spécifier, si nécessaire, l'option "FOR \*" pour lire la totalité du fichier;

soit, de manière plus générale, faire appel à la commande DEFAULTS qui permet de modifier des options par défaut de certaines commandes CMS. En particulier,

```
DEFAULTS SET PEEK FOR *
```

change la valeur par défaut de l'option FOR de la commande PEEK et ce, de façon durable, c'est-à-dire pour la session en cours mais aussi pour les sessions ultérieures, jusqu'à tout nouveau changement.

### 5.1.3.9 Lister et manipuler le contenu du reader: RDRLIST

La commande PEEK est particulièrement intéressante lorsqu'elle est utilisée avec la commande RDRLIST qui permet d'afficher la liste complète des fichiers situés sur votre reader virtuel sous la forme suivante: <sup>7</sup>

```
UZ11101 RDRLIST A0 V 108 TRUNC=108 SIZE=8 LINE=1 COL=1
CMD FILENAME FTYP CLASS USER      NODE      H RCS  DATE
PP          NOTE PUN  A  PP        BLIULG11 N   6 08/21...
UZ36100    NOTE PUN  A  UZ36100    BLIULG11 N   5 08/22...
A          A    PUN  A  PP        BLIULG11 N  11 08/22...
none      none CON  T  PP        BLIULG11 U 121 08/23...
none      none CON  T  PP        BLIULG11 U 449 08/23...
CSM       DATA PUN  A  UZ33100    BLIULG11 N   91 08/23...
```

```
1=Help 2=Refresh 3=Quit      4=Stype 5=Sdate 6=Suser
7=Back 8=Forward 9=Receive 10=      11=Peek 12=Cursor
```

où l'on retrouve pour chaque fichier situé sur le reader (6 dans notre exemple):

1. le filename et le filetype du fichier
2. son type et sa classe
3. le userid du créateur
4. l'ordinateur d'origine (node)
5. s'il est holdé (U) ou non (N)
6. le nombre d'enregistrements
7. la date et l'heure de création

Il suffit alors de positionner le curseur sur la ligne reprenant le fichier que vous désirez visualiser et d'enfoncer la touche FP11 qui provoquera un PEEK de ce fichier ou FP9 pour le copier sur votre minidisque A.

Le bas de l'écran affiché par RDRLIST donne la liste des commandes associées aux autres touches fonctions.

### 5.1.3.10 FILEDEF

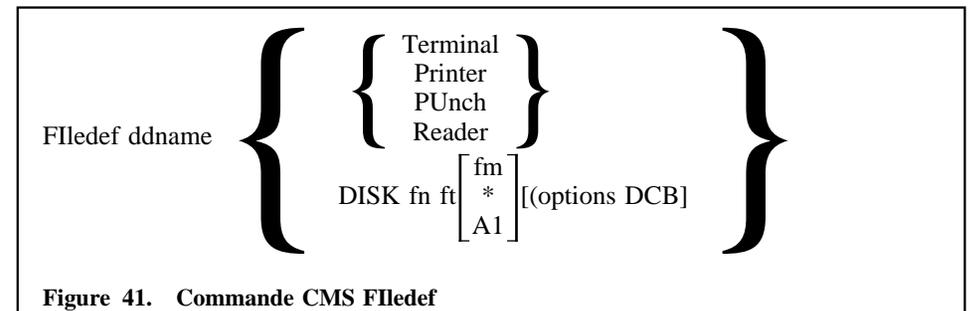


Figure 41. Commande CMS FILEDEF

Remarques:

<sup>7</sup> Cette présentation s'applique uniquement aux terminaux fonctionnant en mode 3270. Sur les télétypes on reçoit la présentation ligne à ligne prévue par XEDIT sur ce genre de matériel (Voir «L'édition en CMS» à la page 25) qui donne accès aux mêmes fonctions en faisant "EXECUTE nom de la fonction" pour agir sur le fichier indiqué par la ligne courante.

- Le fm A est pris par défaut.
- Le fm \* indique l'ordre de balayage habituel.

Cette commande permet de spécifier le fichier ou le UR visé lorsqu'on lit ou écrit en utilisant le ddname 'ddname'.

Les options DCB sont obligatoires si elles sont omises dans le programme. Elles doivent être codifiées de la manière suivante :

```
FI ddname DISK fn ft fm (LRECL 1 BLOCK b RECFM [
  F
  FB
  V
  VB
  VBS
  VS
  U
])
```

**Figure 42. Commande CMS FILEDEF**

Avec VBS et VS il faut utiliser le filemode number 4

### 5.1.3.11 Copie par ddname :

```
MOVEfile [ { ddname1 } [ { ddname2 } ] ]
          [ INMOVE ] [ OUTMOVE ] ]
```

**Figure 43. Commande CMS MOVEFILE**

Remarque: INMOVE et OUTMOVE sont pris par défaut.

Cette commande permet de copier le fichier associé à ddname1 dans le fichier associé à ddname2.

Exemple: lister au terminal le fichier situé en tête du reader:

```
FI INMOVE R
FI OUTMOVE T
MOVE
```

### 5.1.3.12 Trier un fichier:

```
SORT fn1 ft1 fm1 fn2 ft2 fm2
```

**Figure 44. Commande CMS SORT**

permet de trier le fichier fn1 ft1 fm1 (qui doit contenir des enregistrements de longueur fixe) dans l'ordre EBCDIC ascendant et de ranger le résultat du tri dans le fichier fn2 ft2 fm2.

La commande SORT donne le message ENTER SORT FIELD auquel on répond en donnant une ou plusieurs paires de nombres de la forme "XX YY" indiquant la position de début et de fin de(s) zone(s) de tri.

Exemple :

```
SORT TEST DATA A TESTS DATA A
enter sort field
1 3 5 6
```

ce qui signifie trier TEST DATA A sur les bytes 1 à 3 et 5 à 6 et ranger le résultat dans TESTS DATA A.

Remarque : un second programme de tri/fusion plus élaboré est disponible sous CMS. Ce programme est un produit de la firme Computer Associates et s'appelle CA-SORT. Il permet le tri et la fusion d'enregistrements de longueur fixe ou variable quelque soit le fichier support (fichier CMS, fichier VSAM ou bande magnétique). Il accepte également plusieurs types d'interaction de la part du programme au niveau de la logique interne et ce via des EXITS. Une brochure intitulée "Reference Guide CA-SORT CMS" est disponible au bureau de documentation du SEGI.

## 5.1.4 Gérer les bibliothèques MACLIB, TXTLIB et LOADLIB

Les commandes CMS MACLIB, MACLIST, TXTLIB et LOADLIB permettent de gérer les bibliothèques de macros (MACLIB), de modules objets (TXTLIB) et de load modules (LOADLIB).

### 5.1.4.1 MACLIB :

```
MAClib GEN filename fn1 [fn2...]
MAClib ADD filename fn1 [fn2...]
MAClib REP filename fn1 [fn2...]
MAClib DEL filename mem1 [mem2..]
MAClib COMP filename
MAClib MAP filename [(TERM )
```

**Figure 45. Commande CMS MACLIB**

Avant d'introduire un membre dans une bibliothèque MACLIB, il convient de créer la macro dans un fichier dont le filename sera le nom du membre et le filetype sera COPY.

Ensuite, on exécute la commande CMS MACLIB pour créer la bibliothèque (GEN), ajouter des membres à une bibliothèque existante (ADD) ou remplacer des membres existants (REP).

Exemple: Pour créer une bibliothèque BIBLI MACLIB et y ranger les 3 macros MAC1, MAC2 et MAC3, il faut:

1. créer les fichiers MAC1 COPY, MAC2 COPY, MAC3 COPY
2. faire MAC GEN BIBLI MAC1 MAC2 MAC3

Pour détruire un ou plusieurs membres d'une bibliothèque MACLIB, on utilise l'option DEL. Pour obtenir la liste des membres de la bibliothèque, on utilise l'option MAP qui crée un fichier filename MAP A1 contenant cette liste (sauf si l'on spécifie TERM pour avoir la liste directement au terminal). Après plusieurs REPLACE ou DELETE, on a intérêt à faire un compress (COMP) pour récupérer l'espace réservé aux membres détruits.

#### 5.1.4.2 MACLIST:

La commande MACLIST fournit la liste des membres d'une bibliothèque MACLIB. Dans l'environnement MACLIST, on peut taper des commandes en face de chaque membre de la liste.

```
{ MACLIST } nom-de-librairie
  ML
```

Figure 46. Commande CMS MACLIST

En fait, quand on utilise la commande MACLIST, on est placé dans l'environnement XEDIT et on édite le fichier userid MACLIST A0.

Les concepts sont similaires à la commande FILELIST décrite plus loin (voir «Généralisation des manipulations de fichiers sur 3270: FILELIST» à la page 24).

#### 5.1.4.3 TXTLIB :

```
TXTlib GEN filename fn1 [fn2 ...]
TXTlib ADD filename fn1 [fn2 ...]
TXTlib DEL filename mem1[mem2 ...]
TXTlib MAP filename [(TERM)]
```

Figure 47. Commande CMS TXTLIB

Avant d'introduire un module objet dans une bibliothèque TXTLIB, il convient de disposer de ce membre dans un fichier dont le filetype sera TEXT.

On exécute alors la commande CMS TXTLIB pour créer la bibliothèque (GEN) ou ajouter des membres à une bibliothèque existante (ADD): pour remplacer un membre, il faut détruire l'ancien (DEL) puis ajouter le nouveau (ADD).

Exemple:

Pour ajouter le contenu du fichier TEST TEXT dans la bibliothèque BIBLI TXTLIB: TXT ADD BIBLI TEST

Pour obtenir la liste de membres d'une TXTLIB, on utilise l'option MAP qui donnera le résultat dans un fichier filename MAP A5 sauf si l'option TERM a été stipulée, auquel cas le résultat est affiché sur le terminal. MAP donne pour chaque membre son nom (c'est-à-dire le nom de la première CSECT rencontrée dans le fichier TEXT original, son emplacement et la liste de ses points d'entrées (références externes). Si le fichier TEXT original contenait une instruction NAME, le membre portera ce nom uniquement (aucun point d'entrée annexe renseigné, sauf si l'on a spécifié des instructions ALIAS). Le PL/I Optimizer engendre automatiquement une carte NAME dans le fichier fn TEXT A1 lorsqu'on utilise le paramètre NAME du compilateur: ceci évite d'avoir plusieurs noms de CSECT renseignés dans la directory de la bibliothèque TXTLIB (le compilateur PL/I Optimizer produit toujours plusieurs CSECTs dont la première s'appelle systématiquement PLISTART, ce qui, à défaut de carte NAME, amènerait à obtenir une bibliothèque dont tous les membres s'appelleraient PLISTART). En Fortran, pour éviter que tous les programmes principaux (le problème ne se pose pas pour les sous-programmes) ne soient renseignés sous le nom MAIN, il faut également utiliser le paramètre NAME du compilateur. De plus, l'option NAME doit être accompagnée de l'option SYSTEM(CMS).

#### 5.1.4.4 LOADLIB:

```
LOADLIB { { LIST } fid
          { COMPRESS } }
        COPY fid1 fid2 [ (REPLACE
                        MODIFY ) ]
```

Figure 48. Commande CMS LOADLIB

Remarque: fid représente le nom de la bibliothèque sous la forme fn ft fm.

**LIST** permet d'obtenir la liste des membres de la bibliothèque (avant de faire LIST il faut vérifier que le ddname SYSUT1 n'est pas défini sinon le fichier correspondant sera pris en lieu et place de fileid).

**COMPRESS** permet de récupérer l'espace réservé aux membres abandonnés (la bibliothèque est complètement recréée).

**COPY** permet de copier des membres de la LOADLIB fid1 dans fid2.

**REPLACE** signifie que les membres de même nom sont remplacés.

**MODIFY** signifie que les membres de même nom ne sont pas remplacés.

N.B.:

l'option COPY ne fonctionne pas avec des load modules en overlay,  
avec COPY, l'opérateur reçoit le message ENTER: pour l'inviter à donner une ou plusieurs instructions

```
SELECT mem1 mem2 ...  
ou  
EXCLUDE mem1 mem2 ...
```

afin de limiter l'opération de copie aux membres sélectionnés ou non exclus.

Pour créer un nouveau membre d'une loadlib, il faut utiliser la commande LKED qui est décrite dans le chapitre (voir «Programmation sous CMS» à la page 45).

## 5.1.5 Généralisation des manipulations de fichiers sur 3270: FILELIST

La commande FILELIST affiche la liste des informations relatives aux fichiers situés sur des disques accédés. Dans l'environnement FILELIST, les informations que l'on peut obtenir avec la commande LISTFILE sont affichées sous le contrôle de l'éditeur XEDIT. L'on peut alors exécuter des commandes de manipulations de fichiers directement à partir de cette liste.

```
FILEList [fn [ft [fm]]]
```

Figure 49. Commande EXEC FILELIST

**fn** représente le filename du ou des fichiers dont les informations sont demandées. Si on remplace le fn par \* tous les fichiers seront pris en considération. L'on peut également coder un filename composé d'une ou plusieurs suites de caractères comprenant des \* qui indiquent chacun une quelconque suite de caractères. Ainsi

```
filelist *d* data
```

permet d'obtenir la liste des fichiers dont le filetype est DATA et le filename quelconque mais contenant le caractère "d".

Le fn peut également contenir un ou plusieurs caractères % qui indiquent de faire la sélection sur base du fn donné mais en remplaçant chaque % par UN caractère quelconque. Ainsi

```
filelist ab%d data
```

permet la sélection de tous les fichiers dont le filetype est DATA et le filename composé de 4 caractères dont les deux premiers sont "ab" et le dernier "d".

**ft** indique le filetype du ou des fichiers à sélectionner. Ft peut également contenir des \* ou des % à l'image du fn.  
**fm** donne le filemode pour la sélection. Par défaut c'est A. \* indique de sélectionner tous les disques accédés.

Si l'on entre la commande

```
filelist *a* pli* a
```

l'on reçoit par exemple le panneau suivant:

```
UZ11101 FILELIST A0 V 108 TRUNC=108 SIZE=10 LINE=1 COL=1  
Cmd Filename Ftype Fm Fmt Lr Rrds Bs Date Time  
ACCENT PLI A1 F 80 68 2 7/16/84 11/59/06  
CHARACTE PLI A1 F 80 45 1 6/27/84 15/01/16  
COLLLOAD PLI A1 F 80 17 1 11/28/83 17/06/49  
HANOI PLIOPT A1 F 80 11 1 6/07/83 9/02/47  
AJOUT PLI A1 F 80 20 1 9/28/82 9/47/09  
TESTALL PLI A1 F 80 106 3 9/14/82 13/50/14  
MORMAXV PLI A1 F 80 20 1 8/05/82 9/35/24  
MORMAX PLI A1 F 80 8 1 7/09/82 13/56/46  
OSDAR PLI A1 F 80 19 1 3/01/82 15/06/24  
OSDA PLI A1 F 80 175 4 2/04/82 14/36/18
```

```
1=Help 2=Refresh 3=Quit 4=Stype 5=Sdate 6=Ssize  
7=Back 8=Forward 9=FL /n 10= 11=XEDIT 12=Cursor
```

```
====>
```

Où l'on retrouve une ligne descriptive pour chaque fichier sélectionné contenant dans l'ordre

1. une zone commande
2. le filename
3. le filetype
4. le filemode
5. le format des enregistrements
6. la dimension maximale des enregistrements
7. le nombre de blocs occupés sur disque
8. la date de dernière écriture dans le fichier
9. l'heure de dernière écriture dans le fichier

Pour exécuter une commande sur l'un des fichiers sélectionnés il suffit alors de positionner le curseur au début de la ligne qui le concerne et d'entrer la commande choisie dans la zone de commandes. Si cette dernière est trop petite vous pouvez continuer à écrire sur les informations affichées( max 79 caractères). On peut ainsi entrer différentes commandes sur différentes lignes et dès que l'on fera ENTER elles seront toutes exé-

cutées et l'écran sera remis à son état initial tout en tenant compte des modifications effectuées sur la sélection.

Le bas de l'écran indique les commandes associées aux différentes touches de fonctions. Ainsi

- PF1** affiche un HELP
- PF2** rafraîchit la liste en tenant compte des fichiers créés ou effacés depuis le FILELIST.
- PF3** quitte l'environnement FILELIST
- PF4** trie la liste selon le filetype et le filename
- PF5** trie la liste selon la date et l'heure
- PF6** trie la liste selon la taille des fichiers
- PF7** permet de remonter dans la liste
- PF8** permet de descendre dans la liste
- PF9** permet d'afficher une liste reprenant tous les fichiers dont le filename est identique à celui renseigné sur la ligne du curseur.
- PF11** permet d'éditer le fichier indiqué par le curseur
- PF12** permet de positionner le curseur dans la zone input renseignée au bas de l'écran par les caractères =====>, ou, s'il s'y trouvait déjà de le repositionner à son ancienne place dans la liste.

Cette zone input peut recevoir les commandes de XEDIT ou la commande EXECUTE qui a la syntaxe suivante:

```
EXECUTE [CURSOR  
lines] [command]
```

**Figure 50. Commande EXEC EXECUTE**

On utilise EXECUTE pour envoyer des commandes CP ou CMS qui font référence à des fichiers de la liste. Pour cela il suffit de remplacer "command" par la commande à exécuter. "cursor" indique que la commande doit opérer sur le fichier situé sur la ligne indiquée par le curseur. "lines" indique le nombre de lignes sur lesquelles la commande doit opérer à partir de la ligne courante <sup>8</sup> (1 par défaut tandis que \* indique jusqu'à la fin de la liste). Quand on utilise EXECUTE la commande est automatiquement complétée par le filename, le filetype et le filemode du (des) fichier(s) sélectionné(s) sauf si elle contient l'un des codes suivants. Ces codes sont remplacés dans la commande par les informations suivantes:

- / filename, filetype et filemode du fichier
- /n filename du fichier
- /t filetype du fichier

- /m filemode du fichier
- /o rien

Ainsi si l'on fait:

```
UZ11101 FILELIST A0 V 108 TRUNC=108 SIZE=10 LINE=1 COL=1
Cmd Filename Ftype Fm Fmt Lr Rrds Bs Date Time
ACCENT PLI A1 F 80 68 2 7/16/84 11/59/06
CHARACTE PLI A1 F 80 45 1 6/27/84 15/01/16
COLLLOAD PLI A1 F 80 17 1 11/28/83 17/06/49
HANOI PLIOPT A1 F 80 11 1 6/07/83 9/02/47
AJOUT PLI A1 F 80 20 1 9/28/82 9/47/09
TESTALL PLI A1 F 80 106 3 9/14/82 13/50/14
MORMAXV PLI A1 F 80 20 1 8/05/82 9/35/24
MORMAX PLI A1 F 80 8 1 7/09/82 13/56/46
OSDAR PLI A1 F 80 19 1 3/01/82 15/06/24
OSDA PLI A1 F 80 175 4 2/04/82 14/36/18
```

```
1=Help 2=Refresh 3=Quit 4=Stype 5=Sdate 6=Ssize
7=Back 8=Forward 9=FL /n 10= 11=XEDIT 12=Cursor
```

```
=====> execute * compare / /n /t z
```

FILELIST va automatiquement engendrer les commandes suivantes:

```
compare ACCENT PLI A1 ACCENT PLI Z
compare CHARACTE PLI A1 CHARACTE PLI Z
compare COLLLOAD PLI A1 COLLLOAD PLI Z
compare HANOI PLIOPT A1 HANOI PLIOPT Z
compare AJOUT PLI A1 AJOUT PLI Z
compare TESTALL PLI A1 TESTALL PLI Z
compare MORMAXV PLI A1 MORMAXV PLI Z
compare MORMAX PLI A1 MORMAX PLI Z
compare OSDAR PLI A1 OSDAR PLI Z
compare OSDA PLI A1 OSDA PLI Z
```

## 5.2 L'édition en CMS

### 5.2.1 Qu'est-ce qu'un éditeur?

L'éditeur est un programme permettant de créer et de modifier des fichiers situés sur disque. Pour ce faire tout éditeur dispose d'un jeu de commandes réalisant ces différents

<sup>8</sup> La notion de ligne courante est décrite dans le chapitre sur XEDIT. (Voir «L'édition en CMS»)

objectifs. Sa puissance dépendra donc des possibilités offertes par ces commandes ainsi que de la facilité avec laquelle elles peuvent être utilisées.

Disposant d'un important jeu de commandes, l'éditeur XEDIT se recommande également des trois qualités suivantes:

1. il permet d'éditer tout type de fichier sans regard sur leur contenu ou sur leurs caractéristiques (longueur des enregistrements, organisation séquentielle ou directe...).
2. il gère l'écran en "pleine page" (fullscreen), ce qui permet de remplir et/ou de modifier un écran complet avant de l'envoyer vers l'ordinateur d'où un gain sensible au niveau de la performance et un plus grand confort d'utilisation.
3. il permet à chacun de compléter le jeu de commandes standard en écrivant ses propres commandes au moyen d'un macro langage disponible sous CMS et appelé REXX (Voir «Le langage REXX.» à la page 39)

## 5.2.2 Utilisation de XEDIT sur 3270 (DISPLAY MODE)

### 5.2.2.1 Création d'un fichier.

La commande permettant de créer un fichier est:

```
XEDIT filename filetype filemode [(options
```

**Figure 51. Commande CMS XEDIT**

Cette commande fait appel à l'éditeur et définit ainsi le début de la session d'édition. La session d'édition se terminera lorsqu'on entrera une commande signalant à l'éditeur qu'on a fini de manipuler le fichier.

Signalons la possibilité d'éditer un membre d'une MACLIB en spécifiant l'option MEMber nom de membre. Dans ce cas, le filetype doit être obligatoirement MACLIB.

Créons maintenant un fichier contenant un programme FORTRAN de test. nous désignons appeler ce fichier TEST FORTRAN A et donc nous entrons la commande **XEDIT TEST FORTRAN A** puis on enfonce la touche **ENTER** ce qui fera apparaître l'écran suivant:

```
01 TEST FORTRAN A1 F 80 TRUNC=72 SIZE=0 LINE=0 COLUMN=1
02
.
.
10
11 ===== * * * TOP OF FILE * * *
12 !...+...1...+...2...+...3...+...4...+...5..
13 ===== * * * END OF FILE * * *
14
.
.
22
23 ==> _
24
X E D I T 1 FILE
```

Voici ce que ces informations représentent:

- ligne 1** identification du fichier. On y retrouve le filename filetype et filemode du fichier édité. F 80 indique que chaque enregistrement a une longueur fixe de 80 bytes. TRUNC=72 signifie que si l'on entre plus de 72 caractères par ligne celle-ci sera tronquée après la colonne 72. SIZE=0 indique le nombre de lignes dans le fichier (zéro puisque nous n'y avons encore rien mis). LINE=0 est le numéro de la ligne courante(intensité forte).
- ligne 2** ligne des messages. Cette ligne est utilisée par l'éditeur pour afficher un message lorsqu'il détecte une erreur dans la commande que vous venez d'entrer ou pour fournir des informations complémentaires.
- ligne 23** ligne des commandes. C'est sur cette ligne et à la suite des caractères ==> que vous pouvez entrer les commandes permettant de communiquer avec l'éditeur. Après avoir entré une commande il faut pousser sur la touche **ENTER** pour l'envoyer à l'éditeur: à ce moment la commande disparaît et est exécutée.
- ligne 24** zone d'état. Le coin inférieur droit de l'écran donne à tout instant l'état de votre session d'édition, cad EDIT MODE ou INPUT MODE (INPUT MODE est décrit ci-après) ainsi que le nombre de fichiers édités(1 dans notre exemple).
- lignes 11 et 13** zone préfixe. La zone préfixe (====) apparaît à gauche de chaque ligne du fichier. Elle est destinée à recevoir des **PREFIX COMMANDS** dont nous discuterons par la suite.
- lignes 3 à 10** constituent la zone DATA disponible pour afficher le contenu du fichier édité. Les lignes du fichier seront insérées dans cette zone entre le **\* \* \* TOP OF FILE \* \* \*** et le **\* \* \* END OF FILE \* \* \***.
- lignes 14 à 22** idem que lignes 3 à 10.
- ligne 11** ligne courante. La ligne courante est la ligne située au milieu de l'écran (au-dessus de l'échelle repère). Elle apparaît en luminosité forte. Cette notion de ligne courante est fort importante étant donné que plusieurs commandes de l'éditeur remplissent leur fonction à partir de cette ligne. Evidemment vous pouvez changer la position de cette ligne courante. Il faut encore noter que plusieurs commandes

exercer leur action à partir de cette ligne courante et changent sa position lorsqu'elles se terminent.

**ligne 12** l'échelle repère (SCALE). Elle apparaît sous la ligne courante et permet de repérer plus aisément les marges sur lesquelles on travaille.

### 5.2.2.2 Encodage des données.

La première manière d'entrer des données est d'utiliser le mode INPUT. Pour ce faire il faut entrer la commande

```
INPUT
(====> INPUT suivi de ENTER).
```

**Figure 52. Commande XEDIT INPUT**

On obtient alors l'écran suivant:

```
01 TEST FORTRAN A1 F 80 TRUNC=72 SIZE=9 LINE=0 COLUMN=1
02 INPUT MODE:
03
.
.
11
12 * * * TOP OF FILE * * *
13 !...+...1...+...2...+...3...+...4...+...5..
14 _
15
.
.
22
23 ====> * * * input zone * * *
24                                     INPUT-MODE 1 FILE
```

En mode input, la zone préfixe disparaît (====). Les zones message et état affichent le libellé **INPUT-MODE**. La ligne commande contient le libellé **\* \* \* INPUT ZONE \* \* \***, ce qui signifie qu'on **ne peut pas entrer de commandes** dans cette zone en mode input. Le curseur est placé au début de la première ligne destinée à recevoir vos données. Après avoir entré une ligne vous enfoncez la touche NEWLINE pour positionner le curseur au début de la ligne suivante et ainsi continuer en entrant la seconde ligne de données. Entrons donc les trois lignes suivantes:

```
do 10 i=1,10
10 print 100,i
100 format(1h1,i2)
```

Après avoir enfoncé la touche ENTER l'écran apparaîtra comme suit:

```
01 TEST FORTRAN A1 F 80 TRUNC=72 SIZE=12 LINE=3 COLUMN=1
02 INPUT MODE:
03
.
.
08
09 * * * TOP OF FILE * * *
10      DO 10 I=1,10
11 10    PRINT 100,I
12 100   FORMAT(1H1,I2)
13 !...+...1...+...2...+...3...+...4...+...5..
14 _
15
.
.
22
23 ====> * * * INPUT ZONE * * *
24                                     INPUT-MODE 1 FILE
```

Les lignes que nous venons d'entrer ont été déplacées dans la moitié supérieure de l'écran. La dernière ligne entrée (100 format(1h1,i2)) est devenue la ligne courante et toutes les lettres sont converties en majuscules. Pour compléter le fichier il suffit de répéter cette procédure. Si vous n'avez plus de lignes à entrer, il suffit d'enfoncer une fois encore la touche ENTER. Ceci clôture le mode INPUT et vous revenez en mode EDIT. L'écran apparaît alors comme suit:

```
01 TEST FORTRAN A1 F 80 TRUNC=72 SIZE=3 LINE=3 COLUMN=1
02 XEDIT:
03
.
.
08
09 ===== * * * TOP OF FILE * * *
10 ===== DO 10 I=1,10
11 ===== 10 PRINT 100,I
12 ===== 100 FORMAT(1H1,I2)
13          !...+...1...+...2...+...3...+...4...+...
14 ===== * * * END OF FILE * * *
15
.
.
22
23 ====> _
24                                     X E D I T 1 FILE
```

L'écran est donc replacé en mode EDIT et le fichier contient vos données. Notez au passage que la ligne identification renseigne qu'il y a trois lignes dans le fichier (SIZE=3) et que la ligne courante est la troisième (LINE=3).

On peut revenir en mode INPUT à l'aide de la commande **INPUT** pour entrer de nouvelles données **après la ligne courante** puis revenir en mode EDIT en faisant deux ENTERs successifs et ainsi de suite. Nous verrons par la suite comment insérer des lignes entre des lignes existantes.

Lorsqu'on a fini de travailler avec ce fichier il convient de dire à l'éditeur ce qu'il doit en faire. En effet actuellement le fichier est en mémoire virtuelle et non sur disque. Pour le mettre sur disque il faut entrer la commande

```
FILE
```

**Figure 53. Commande XEDIT FILE**

Cette commande FILE signale à l'éditeur de sauver le fichier sur disque et de clôturer la session d'édition en cours.

**Caractères minuscules et majuscules:** Dans le premier exemple que nous venons de réaliser, toutes les lettres ont été converties par l'éditeur en majuscules. Pour signaler que l'on désire maintenir la distinction minuscules/majuscules, il faudra utiliser la commande

```
SET CASE MIXED
```

**Figure 54. Commande XEDIT SET CASE**

Dès la lecture de cette commande, l'éditeur préservera donc le format des caractères que vous entrez au terminal.

**Eclatement et jointure des lignes:** pour éclater ou joindre des lignes on utilisera une PFK en lieu et place d'une commande suivie d'un ENTER. Il faut en effet savoir que XEDIT affecte à chaque PFK une signification particulière: il en résulte que lorsqu'on enfonce l'une des PFK, la fonction qui lui est associée est exécutée sans que l'on soit obligé d'enfoncer également la touche ENTER.

Remarque: dans ces notes nous nous limiterons aux PFK 1 à 12. Si votre terminal possède 24 PFK, les PFK 13 à 24 ont la même signification que les PFK 1 à 12.

Pour éclater une ligne il suffit de procéder comme suit:

- positionner le curseur sous le caractère à partir duquel on désire effectuer l'éclatement.

- enfoncez la PF11.

pour joindre deux lignes on procédera comme suit:

- positionner le curseur sur la ligne où vous désirez la jointure.
- enfoncez la touche PF11.

Si le curseur est en fin de ligne, PF11 provoquera une jointure, sinon ce sera un éclatement.

**La tabulation:** dans certains cas vous désirez placer vos données dans des colonnes bien précises. Pour cela la touche PF4 fonctionne comme une touche de tabulation. Chaque fois que vous enfoncez cette touche, le curseur se déplacera vers la colonne de tabulation suivante. La tabulation initiale est fixée par l'éditeur en fonction du filetype du fichier édité. Vous pouvez changer cette définition au moyen de la commande **SET TABS**.

Vous pouvez lister à tout instant la définition courante de la tabulation en entrant la commande **QUERY TABS**. Pour remplacer cette définition courante vous entrez la commande

```
SET TABS n1[n2[n3...]]
```

**Figure 55. Commande XEDIT SET TABS**

Ainsi si nous désirons créer le fichier TEST DATA A devant contenir des valeurs dans les colonnes 1, 10, 20, 30 et 40, on fera la commande SET TABS 1 10 20 30 40 suivie de la commande INPUT. Ensuite, après avoir entré chaque valeur on enfoncez la touche PF4 afin que le curseur se déplace au début de la zone suivante.

**Affichage des fonctions associées aux PFK:** pour obtenir la liste des fonctions associées aux pfk il suffit d'entrer la commande

```
QUERY PF
```

**Figure 56. Commande XEDIT QUERY PF**

**Insertion de mots dans une ligne existante:** imaginons que nous ayons créé le fichier TEST FORTRAN A comme suit:

```

01 TEST FORTRAN A1 F 80 TRUNC=72 SIZE=3 LINE=3 COLUMN=1
02 XEDIT:
03
.
.
08
09 ===== * * * TOP OF FILE * * *
10 ===== DO 10 I=1,10
11 ===== 10 PRINT 100,I
12 ===== 100 FORMAT(1H ,I2)
13 !...+...1...+...2...+...3...+...4...+
14 ===== * * * END OF FILE * * *
15
.
.
22
23 ==> _
24 X E D I T 1 FILE

```

Pour modifier le format afin d'obtenir "FORMAT (1H ,I vaut ',I2)" on procédera comme suit:

- positionner le curseur à l'endroit d'insertion
- enfoncer la touche **PA2**(ou AP2 selon le type de clavier).
- enfoncer la touche **INSERT MODE**
- entrer les caractères 'I vaut'
- enfoncer la touche **RESET** (ou RESTAU selon le type de clavier) pour quitter le mode insertion.

## Résumé

Commandes/PFK	Fonction
<b>XEDIT</b>	créer un fichier
<b>INPUT</b>	passer en mode INPUT pour entrer les données
<b>FILE</b>	écrire le fichier sur disque
<b>QUERY TABS</b>	afficher la tabulation
<b>SET TABS</b>	changer la tabulation
<b>SET CASE MIXED</b>	accepter minuscules/majuscules
<b>PF4</b>	touche de tabulation
<b>PF11</b>	jointure ou éclatement de deux lignes
<b>PA2 + MODE INS</b>	insertion dans une ligne
<b>QUERY PF</b>	afficher les fonctions PF
<b>QUERY PFn</b>	afficher la fonction PFn
<b>RESET</b>	sortir du mode insertion

### 5.2.2.3 Modifier un fichier (Première partie).

Une fois le fichier créé vous serez probablement amenés à devoir le modifier. Cette section explique comment lister le fichier, ajouter des lignes ou encore supprimer, répéter, déplacer ou copier des lignes.

**Afficher le fichier:** pour réutiliser une copie du fichier située sur disque, on utilise la même commande XEDIT comme pour créer ce fichier. Cette commande provoque l'affichage du fichier, ce dernier restant inchangé sur disque tant que l'on n'aura pas entré une commande FILE.

**On a oublié le nom du fichier...:** si vous avez oublié le nom du fichier que vous voulez afficher, vous faites appel à la commande **FILEL** (voir «Généralisation des manipulations de fichiers sur 3270: FILELIST» à la page 24)

**Ajouter des lignes:** vous voulez compléter votre programme TEST en y ajoutant des lignes. Une première méthode que nous avons déjà vue est d'utiliser la commande INPUT. : Une autre méthode consiste à utiliser une commande préfixe qui ajoute des lignes blanches dans le fichier. Une commande préfixe est toujours codée dans la zone préfixe. Une fois les lignes blanches ajoutées, il suffit de les remplir avec vos nouvelles données.

Vous entrez la commande préfixe n'importe où dans la zone préfixe constituée des cinq signes (= =====). La commande préfixe peut être entrée soit en majuscules, soit en minuscules et doit être suivie d'un enfoncement de la touche ENTER.

La commande préfixe qui ajoute une ligne blanche après la ligne sur laquelle elle a été codée est **A**. Vous pouvez entrer un nombre devant ou derrière la commande A pour insérer plus d'une ligne blanche. Exemple:

```

01 TEST FORTRAN A1 F 80 TRUNC=72 SIZE=3 LINE=3 COLUMN=1
02 XEDIT:
03
.
.
08
09 ==A== * * * TOP OF FILE * * *
10 ==2A= DO 10 I=1,10
11 ===== 10 PRINT 100,I
12 ==A== 100 FORMAT(1H ,I2)
13 !...+...1...+...2...+...3...+...4...+
14 ===== * * * END OF FILE * * *
15
.
.
22
23 ==> _
24 X E D I T 1 FILE

```

Après avoir enfoncé la touche ENTER on obtiendra l'écran suivant:

```
01 TEST FORTRAN A1 F 80 TRUNC=72 SIZE=6  LINE=6 COLUMN=1
02 XEDIT:
03
04
05 ===== * * * TOP OF FILE * * *
06 =====
07 ===== DO 10 I=1,10
08 =====
09 =====
10 ===== 10 PRINT 100,I
11 ===== 100 FORMAT(1H ,I2)
12 =====
13 !...+...1...+...2...+...3...+...4...+.
14 ===== * * * END OF FILE * * *
15
.
.
22
23 ==> _
24
X E D I T 1 FILE
```

Cela étant fait il suffit donc de compléter les lignes blanches puis de faire la commande FILE pour sauver le fichier sur disque.

Si on veut insérer de nouvelles lignes dans un texte structuré on peut utiliser la commande préfixe SI ("Structured Input"). Il suffit d'entrer la commande préfixe SI dans la zone préfixe de la ligne après laquelle on veut ajouter des lignes. Tant qu'on n'entre pas une ligne vide, on pourra insérer du texte le curseur étant positionné sur la même colonne que pour la ligne de départ.

Une dernière méthode consiste à positionner le curseur sur la ligne après laquelle vous désirez insérer une ligne blanche puis d'enfoncer la touche PF2. En effet cette touche est associée par défaut à la commande SOS LINEAdd qui remplit cette fonction.

**Balayer le fichier:** si l'on désire continuer à ajouter des lignes au fichier et qu'il n'y a plus de place sur l'écran, on devra balayer le fichier de la même manière que l'on tournerait les pages d'un livre. Pour ce faire on dispose des commandes de balayage.

**Balayage avant et arrière:** deux PFK permettent de balayer le fichier: **PF8** permet de passer à la page suivante et **PF7** permet de revenir à la page précédente. Pour tourner plusieurs pages il suffit d'appuyer plusieurs fois sur ces touches.

**Revenir en fin ou début du fichier:** si le fichier est volumineux, plutôt que d'utiliser plusieurs fois PF8 pour se placer sur la dernière page il suffit d'entrer la commande

BOTTOM

Figure 57. Commande XEDIT BOTTOM

De même pour revenir à la première page du fichier sans devoir utiliser plusieurs fois la commande PF7, il suffit d'entrer la commande

TOP

Figure 58. Commande XEDIT TOP

**Balayage ligne à ligne:** pour déplacer la fenêtre d'un nombre de lignes spécifié plutôt que par écrans entiers, on utilisera les commandes

DOWN n

Figure 59. Commande XEDIT DOWN

UP n

Figure 60. Commande XEDIT UP

La commande DOWN n déplace la fenêtre de n lignes vers le bas tandis que la commande UP n déplace la fenêtre de n lignes vers le haut.

Une autre manière de définir une nouvelle ligne courante consiste à placer un SLASH dans la zone préfixe de la nouvelle ligne courante et d'enfoncer la touche ENTER.

**Supprimer des lignes** la commande préfixe **D** supprime la ligne sur laquelle le D a été entré (ne pas confondre avec la touche DEL ou 'IMPR SUPP' qui supprime des caractères et des blancs à l'intérieur d'une ligne). :exemples:

```
====d supprime cette ligne
d==== supprime cette ligne
10d== supprime cette ligne et les neuf suivantes
```

Lorsque vous enfonchez la touche ENTER les lignes affectées par D disparaissent.

**Supprimer un bloc de lignes:** un bloc de lignes est un ensemble de lignes consécutives. Pour supprimer un bloc sans devoir compter les lignes il suffit de mettre une commande préfixe **DD** sur la première ligne du bloc et une autre commande DD sur la dernière ligne du bloc à supprimer. :Si vous entrez DD sur une zone préfixe puis que vous utilisez PF7 ou PF8 pour balayer le fichier, la zone d'état affichera le message "BLOCK INCOMPLETE". Ce message indique simplement que l'éditeur attend que vous entriez le second DD pour délimiter le bloc à supprimer.

**Récupérer des lignes supprimées:** si, par mégarde, vous avez supprimé des lignes utiles au moyen des commandes D ou DD, vous pouvez durant une même session d'édition les récupérer au moyen de la commande

```
RECOVER n
```

Figure 61. Commande XEDIT RECOVER

La commande RECOVER n insère les n dernières lignes supprimées devant la ligne courante.

**Répéter des lignes:** pour répéter une ligne pour pouvez utiliser la commande préfixe " (double apostrophe). Pour répéter n fois la même ligne, il suffit de faire n" ou "n. : Exemple:

```
3"==== JE VEUX 3 COPIES SUPPLEMENTAIRES DE CETTE LIGNE
==== OH OUI ?
```

donnera:

```
==== JE VEUX 3 COPIES SUPPLEMENTAIRES DE CETTE LIGNE
==== JE VEUX 3 COPIES SUPPLEMENTAIRES DE CETTE LIGNE
==== JE VEUX 3 COPIES SUPPLEMENTAIRES DE CETTE LIGNE
==== JE VEUX 3 COPIES SUPPLEMENTAIRES DE CETTE LIGNE
==== OH OUI ?
```

**Copier des lignes:** supposons que vous ayez une partie du programme édité qui doit être recopié à un autre endroit du fichier. Pour éviter de le coder à nouveau vous pouvez utiliser la commande préfixe C. :Pour copier une ligne on entre donc un C dans la zone préfixe de la ligne puis un F(F=FOLLOWING) ou un P(P=PRECEDING) respectivement dans la zone préfixe de la ligne qui précède ou qui suit l'endroit de destination. Après avoir enfoncé la touche ENTER vous verrez que la ligne a été recopiée à l'endroit spécifié. Exemple:

```
====c CECI EST LA LIGNE QUE JE VEUX COPIER
.
.
====f JE VEUX LA COPIER APRES CETTE LIGNE

Après avoir enfoncé la touche ENTER on aura

==== CECI EST LA LIGNE QUE JE VEUX COPIER
.
.
==== JE VEUX LA COPIER APRES CETTE LIGNE
==== CECI EST LA LIGNE QUE JE VEUX COPIER
```

**Copier plusieurs lignes:** vous pouvez faire précéder ou suivre la commande préfixe C d'un nombre indiquant le nombre de lignes participant à l'opération de copie.

**Copier un bloc de lignes:** vous pouvez également définir un bloc de lignes participant à l'opération de copie. Pour cela on met une commande préfixe CC. au début et à la fin du bloc (comme pour la suppression d'un bloc) et vous indiquez la destination au moyen des commandes préfixes F ou P. Après avoir enfoncé la touche ENTER, le bloc sera recopié.

Si vous balayez le fichier avant d'avoir entré la deuxième commande préfixe CC, vous recevrez l'un des messages "COPY/MOVE PENDING" ou "BLOCK INCOMPLETE". Ceci signifie que l'éditeur attend que vous entriez soit le F ou le P, soit le second CC.

**Déplacer des lignes:** le déplacement de lignes signifie que l'on désire retirer les lignes de leur endroit initial pour les replacer ailleurs. :Pour cela on utilise les commandes préfixes M ou MM à l'image des commandes préfixes C et CC.

**Restauration:** lorsqu'on veut annuler une demande de copie ou de déplacement et tant que l'on a l'un des messages "BLOCK INCOMPLETE" ou "COPY/MOVE PENDING", on peut entrer la commande

```
RESET
```

Figure 62. Commande XEDIT RESET

pour rejeter l'opération demandée (ne pas confondre la commande ==>RESET avec la touche RESET).

## Résumé.

### Commandes/pfk Fonction

<b>FILEList</b>	affiche la liste des fichiers
<b>BOTTOM</b>	ligne courante = dernière ligne
<b>TOP</b>	ligne courante = première ligne
<b>DOWN</b>	descendre la ligne courante
<b>UP</b>	monter la ligne courante
<b>RECOVER</b>	récupérer des lignes supprimées
<b>RESET</b>	annuler un copiage ou déplacement
<b>A(prefixe)</b>	ajout de lignes blanches
<b>/ (prefixe)</b>	définit la nouvelle ligne courante
<b>D(prefixe)</b>	supprime une ligne(s)
<b>"(prefixe)</b>	répète une ligne(s)
<b>C(prefixe)</b>	copie une ligne(s)
<b>M(prefixe)</b>	déplace une ligne(s)
<b>F(prefixe)</b>	C/M après cette ligne
<b>P(prefixe)</b>	C/M avant cette ligne
<b>PF2</b>	insérer une ligne blanche
<b>PF7</b>	balayage vers le haut
<b>PF8</b>	balayage vers le bas
<b>SI(préfixe)</b>	entrée de données structurées
<b>LE n</b>	n colonnes vers la gauche
<b>RI n</b>	n colonnes vers la droite
<b>&lt;n(préfixe)</b>	n colonnes vers la gauche
<b>&gt;n(préfixe)</b>	n colonnes vers la droite

### 5.2.2.4 Modifier un fichier (Seconde partie).

Une fois le fichier créé il n'est pas toujours aisé d'y retrouver certaines informations ou encore de modifier une même information qui se retrouve à plusieurs endroits dans le fichier. L'éditeur XEDIT apporte une solution à ces deux situations via les commandes

```
Locate /string/
```

**Figure 63. Commande XEDIT LOCATE**

```
Change /string1/string2/ [ n1 [ n2 ] ]
```

**Figure 64. Commande XEDIT CHANGE**

**Les commandes LOCATE et CHANGE :** Imaginons que le fichier TEST DATA A contienne les informations suivantes:

```
01 TEST DATA A1 F 80 TRUNC=80 SIZE=9 LINE=1 COLUMN=1
02 XEDIT:
03
.
.
10
11 ===== * * * TOP OF FILE * * *
12 ===== AMATO M. R.JONRUELLE 4 5020 LIEGE
13          !...+....1...+....2...+....3...+....4...+.
14 ===== BEER A. R.ST GILLES 9 5000 LIEGE
15 ===== CAYET R. R.CAMPINE 7 5000 LIEGE
16 ===== DURIEUX P. R.THIRIARD 1 5020 LIEGE
17 ===== ENDRES M. R.JAURES 11 5020 LIEGE
18 ===== GEUNS J.L. R.NANIOT 6 5000 LIEGE
19 ===== HAVART J. R.GODET 12 5000 LIEGE
20 ===== IPPOLITO C. R.LULAY 9 5020 LIEGE
21 ===== JANSEN F. R.BODSON 10 5000 LIEGE
22 ===== * * * END OF FILE * * *
23 ==> _
24                                     X E D I T 1 FILE
```

Si l'on désire changer 5000 en 4000 et 5020 en 4020, on peut positionner le curseur sur la zone à modifier puis entrer la correction.

Une autre méthode consiste à utiliser la commande `LOCATE /5000`. vous devez mettre un séparateur entre le nom de la commande et les données à chercher. Dans notre exemple on a choisi le SLASH(/) comme séparateur. Si les données à chercher contiennent un / on peut choisir un autre caractère spécial comme séparateur. Exemple

```
LOCATE $5000
```

```
LOCATE %5000
```

Les caractères ^ (NOT), +(plus), -(moins) et .(point) ne peuvent être utilisés comme séparateurs.

Donc si l'on entre la commande `LOCATE/5000`, l'éditeur va parcourir le fichier à partir de la fin de la ligne courante jusqu'à ce qu'il trouve une ligne contenant le libellé 5000. cette dernière deviendra la nouvelle ligne courante.

Notons qu'une fois le mot trouvé on peut utiliser la commande `CHANGE` pour le modifier en procédant comme suit:

```
CHANGE /5000/4000/
```

Attention la commande `CHANGE` ainsi codée changera le premier mot 5000 dans la ligne courante. C'est pourquoi il faut alors réutiliser la commande `LOCATE` pour définir la ligne contenant le prochain mot 5000 comme nouvelle ligne courante et ainsi de suite....

Lorsque la commande LOCATE ne trouve pas le mot demandé, elle renvoie le message "TARGET NOT FOUND".

**Changer un mot à travers tout le fichier:** il est également possible de changer un mot d'un coup sec à travers tout le fichier. Ainsi si nous entrons la commande

```
CHANGE /5000/4000/ * *
```

l'éditeur nous affichera le résultat suivant:

```
01 TEST DATA A1 F 80 TRUNC=80 SIZE=9 LINE=10 COLUMN=1
02 5 OCCURENCE(S) CHANGED ON 5 LINES
03 ===== AMATO M.      R.JONRUELLE 4 5020 LIEGE
04 ===== BEER A.      R.ST GILLES 9 4000 LIEGE
05 ===== CAYET R.     R.CAMPINE 7 4000 LIEGE
06 ===== DURIEUX P.   R.THIRIARD 1 5020 LIEGE
07 ===== ENDRES M.   R.JAURES 11 5020 LIEGE
08 ===== GEUNS J.L.  R.NANIOT 6 4000 LIEGE
09 ===== HAVART J.   R.GODET 12 4000 LIEGE
10 ===== IPPOLITO C. R.LULAY 9 5020 LIEGE
11 ===== JANSEN F.   R.BODSON 10 4000 LIEGE
12 ===== * * * END OF FILE * * *
13      !...+....1...+....2...+....3...+....4...+.
14
.
.
22
23 ==> _
24
X E D I T 1 FILE
```

Notons ici que le premier astérisque de la commande CHANGE indique que les changements doivent être effectués sur toutes les lignes qui suivent la ligne courante tandis que le second astérisque indique que si le mot à changer apparaît plusieurs fois sur une même ligne il faut le changer chaque fois. Après le changement, la dernière ligne du fichier devient la ligne courante. Donc pour changer 5020 en 4020 que devons nous faire ensuite ?

**Recherche arrière:** la commande LOCATE permet également de faire une recherche en marche arrière. Pour cela il suffit de faire précéder le séparateur d'un signe moins(-). exemple:

```
LOCATE -/5020
```

**Recherche de deux mots adjacents:** pour rechercher deux mots adjacents au moyen de la commande LOCATE, nous devons spécifier également le séparateur de fin comme suit

```
LOCATE /beer a./
```

**La commande =:** pour éviter d'entrer successivement la même commande on peut utiliser la commande = ou la touche **PF9** pour signaler à l'éditeur que l'on veut répéter la dernière commande entrée.

**Résumé.**

**Commandes/PFK Fonction**

<b>LOCATE</b>	retrouver un mot
<b>CHANGE</b>	changer un mot
<b>PF9(=)</b>	répéter une commande

### 5.2.2.5 Modifier un fichier (Troisième partie).

Dans cette section nous verrons comment combiner plusieurs fichiers ou encore trier les lignes du fichier édité.

**Insérer des données d'un autre fichier:** pour insérer une partie ou l'entièreté d'un autre fichier dans le fichier édité on utilisera la commande **GET**.

**Insérer un fichier entier:** la commande GET insère un fichier après la ligne courante du fichier édité. Il faut donc au préalable définir la ligne courante en conséquence ou, si l'on désire insérer les données en début ou en fin de fichier, utiliser au préalable les commandes TOP ou BOTTOM. La commande GET a la syntaxe suivante

```
GET filename filetype [first-line [number-of-lines]]
```

**Figure 65. Commande XEDIT GET**

En réponse à l'insertion on reçoit le message "EOF REACHED" indiquant que tout le fichier a été inséré et la dernière ligne insérée devient la ligne courante. Les deux paramètres facultatifs permettent d'indiquer le numéro de la première ligne à insérer et le nombre de lignes à insérer.

La dernière ligne insérée devient également la nouvelle ligne courante.

**Eviter de sauver les modifications:** si vous ne désirez pas sauver sur disque les modifications apportées au fichier, on sortira de la session d'édition en utilisant la touche **PF3** qui a le même effet que la commande

```
QUIT
```

**Figure 66. Commande XEDIT QUIT**

Si le fichier a été modifié on recevra le message "FILE HAS BEEN CHANGED. USE QUIT TO QUIT ANYWAY". Ce message vous est donné pour vous permettre de rectifier le tir si vous avez malencontreusement appuyé sur la touche PF3. Si donc votre

intention est bien de terminer la session d'édition sans sauver vos modifications il suffit de répondre par lae commande

```
QQuit
```

**Figure 67. Commande XEDIT QQUIT**

La session d'édition est alors terminée et le fichier sur disque n'est pas modifié. Si vous entrez directement la commande QQ, la session d'édition se clôture directement sans même vous demander confirmation et sans sauver vos modifications sur disque.

**Obtenir les numéros des lignes:** certaines commandes, comme par exemple la commande GET, nécessitent de donner les numéros des lignes sur lesquelles elles opèrent. On peut aisément obtenir ces numéros en entrant la commande

```
SET NUMBER ON
```

**Figure 68. Commande XEDIT SET NUMBER ON**

Cette commande va renvoyer dans la zone préfixe les numéros des lignes ce qui n'empêche pas que cette zone reste disponible pour les commandes préfixes. On peut supprimer à tout instant ces numéros en entrant la commande

```
SET NUMBER OFF
```

**Figure 69. Commande XEDIT SET NUMBER OFF**

Ainsi si l'on édite le fichier TEST DATA puis que l'on entre la commande SET NUMBER ON, on obtiendra ceci:

```
01 TEST DATA  A1 F 80 TRUNC=80 SIZE=9 LINE=0  COLUMN=1
02 XEDIT
03
.
.
11
12 00000 * * * TOP OF FILE * * *
13      !...+....1...+....2...+....3...+....4...+.
14 00001 AMATO M.      R.JONRUELLE 4 4020 LIEGE
15 00002 BEER A.      R.ST GILLES 9 4000 LIEGE
16 00003 CAYET R.     R.CAMPINE 7 4000 LIEGE
17 00004 DURIEUX P.  R.THIRIARD 1 4020 LIEGE
18 00005 ENDRES M.   R.JAURES 11 4020 LIEGE
19 00006 GEUNS J.L.  R.NANIOT 6 4000 LIEGE
20 00007 HAVART J.   R.GODET 12 4000 LIEGE
21 00008 IPPOLITO C. R.LULAY 9 4020 LIEGE
22 00009 JANSEN F.   R.BODSON 10 4000 LIEGE
23 ===> _
24                                     X E D I T 1 FILE
```

**Trier un fichier:** la commande **SORT** permet de trier les lignes d'un fichier en se basant sur l'ordre alphanumérique de zones de tri. Cette commande opère à partir de la ligne courante et sur un nombre de lignes renseigné en paramètre. La zone de tri est fixée en termes de numéro colonne début et numéro colonne fin. La syntaxe de la commande SORT est la suivante:

```
SORT number-of-lines column1 column2
```

**Figure 70. Commande XEDIT SORT**

Le nombre de lignes peut être indiqué sous forme d'un astérisque pour signifier que le tri doit être effectué jusqu'à la fin du fichier.

Ainsi dans le fichier TEST DATA A, la rue est indiquée dans les colonnes 15 à 23. Pour trier sur cette zone de tri on fera donc

```
SORT * 15 23
```

ce qui nous donnera le résultat suivant:

```

01 TEST DATA A1 F 80 TRUNC=80 SIZE=9 LINE=0 COLUMN=1
02 XEDIT
03
.
.
11
12 ===== * * * TOP OF FILE * * *
13      !...+....1...+....2...+....3...+....4...+.
14 ===== JANSEN F.   R.BODSON   10 4000 LIEGE
15 ===== CAYET R.    R.CAMPINE   7 4000 LIEGE
16 ===== HAVART J.   R.GODET    12 4000 LIEGE
17 ===== ENDRES M.   R.JAURES   11 4020 LIEGE
18 ===== AMATO M.    R.JONRUELLE 4 4020 LIEGE
19 ===== IPPOLITO C. R.LULAY    9 4020 LIEGE
20 ===== GEUNS J.L.  R.NANIOT   6 4000 LIEGE
21 ===== BEER A.     R.ST GILLES 9 4000 LIEGE
22 ===== DURIEUX P.  R.THIRIARD 1 4020 LIEGE
23 =====> _
24
                                X E D I T   1 FILE

```

Pour trier dans l'ordre décroissant on complète la commande SORT comme suit:

```
SORT number-of-lines D column1 column2
```

Remarque: "D" indique un ordre décroissant.

**Appel à l'aide :** si durant la session d'édition vous avez besoin d'utiliser une commande dont vous avez oublié la syntaxe, vous pouvez demander l'aide de l'éditeur en appuyant sur la touche **PF1**. XEDIT vous affichera alors un menu reprenant la liste des commandes. Il suffit alors de positionner le curseur sous la commande qui vous intéresse et d'enfoncer à nouveau la touche PF1. Vous recevrez alors la description de cette commande. Après avoir lu les renseignements qui vous intéressent, vous enfoncez la touche PF12 pour revenir dans la session d'édition.

#### Résumé.

Commandes/PFK	Fonction
GET	insérer un fichier
SET NUMBER ON	afficher les numéros des lignes
SET NUMBER OFF	effacer les numéros des lignes
PF3(QUIT)	fin session édition
QQUIT	fin session édition sans sauvetage
SORT	trier le fichier
PF1(HELP)	afficher syntaxe des commandes

### 5.2.2.6 Sauvetage du fichier.

Durant l'encodage de vos informations, vous désirez probablement vous mettre à l'abri d'un DOWN de l'ordinateur ou d'une coupure de ligne. XEDIT étant un éditeur mémoire les informations entrées ne seront sauvées sur disque que lors de la commande FILE. Pour minimiser les risques vous pouvez utiliser les commandes **SAVE** ou **SET AUTOSAVE**

**Commande SAVE :** durant la session d'édition on peut entrer la commande SAVE qui permet de décharger le fichier édité sur disque. Les modifications jusqu'alors réalisées sont ainsi sauvées. En cas de DOWN, il suffira donc de refaire la commande XEDIT pour le récupérer. Si vous avez fait une commande SAVE durant la session d'édition vous devrez quand même entrer une commande FILE (ou QUIT) pour clôturer celle-ci. Si vous faites SAVE suivi de QUIT, les modifications faites avant d'entrer la commande SAVE ont été sauvées sur disque, mais pas les modifications réalisées après.

**Commande SET AUTOSAVE:** pour éviter de devoir entrer périodiquement la commande SAVE, l'on peut utiliser la commande

```
SET AUTOSAVE n
```

Figure 71. Commande XEDIT SET AUTOSAVE

Cette dernière offre l'avantage de ne devoir être entrée qu'une seule fois durant la session d'édition. Cette commande provoque un sauvetage automatique chaque fois que vous avez entré ou modifié un certain nombre de lignes. Ainsi si vous désirez que le fichier soit sauvé chaque fois que vous avez modifié dix lignes, vous faites la commande

```
SET AUTOSAVE 10
```

A chaque sauvetage automatique, on reçoit le message "AUTOSAVED AS n AUTOSAVE A1" où n est un nombre donné par XEDIT. Le fichier ainsi sauvé ne porte donc pas le même nom que le fichier original: son filename sera le nombre n et son filetype est AUTOSAVE. En cas de DOWN vous pourrez donc récupérer le fichier à l'état du dernier SAVE automatique en faisant la commande XEDIT n AUTOSAVE suivie de la commande FFILE filename filetype pour le sauver sur disque dans le fichier original (FFILE remplit la même fonction que FILE mais permet de sauver un fichier pour remplacer un fichier existant autre que le fichier édité).

La commande

```
QUERY AUTOSAVE
```

Figure 72. Commande XEDIT QUERY AUTOSAVE

renvoie un message de la forme "AUTOSAVE n1; FILENAME n2; ALTERATIONS n3" où

n1 représente le nombre n spécifié dans la commande SET AUTOSAVE.  
n2 représente le filename du fichier sauvé automatiquement  
n3 est le nombre de lignes modifiées depuis le dernier SAVE automatique

## Résumé.

Commandes/PFK	Fonction
SAVE	sauve le fichier sur disque
SET AUTOSAVE	provoque le SAVE automatique
QUERY AUTOSAVE	affiche état AUTOSAVE

### 5.2.2.7 Définir son PROFILE.

On peut créer un fichier **XEDIT PROFILE** qui contiendra une liste de commandes XEDIT qui seront automatiquement exécutées chaque fois que vous entrerez en session d'édition. Ceci constitue par exemple une manière de définir automatiquement un SET AUTOSAVE ou une tabulation. Exemple:

```
SET AUTOSAVE 10
SET TABS 7
```

### 5.2.2.8 Rappel de commandes.

La touche PF6 permet de rappeler successivement les dernières commandes entrées et de les afficher dans la zone commandes. Ceci évite de réencoder plusieurs fois la même chose.

## 5.3 Dialogues entre machines virtuelles.

Dans cette partie nous ferons la distinction entre l'échange de messages, de notes et de fichiers entre machines virtuelles. Si nous savons tous ce qu'est un fichier il importe cependant de préciser la distinction entre note et message.

1. un message est l'objet d'une communication brève, habituellement composée d'une phrase ou deux. C'est en quelque sorte l'information que vous transmettez souvent par téléphone. Vous pouvez utiliser la commande TELL pour envoyer un message (la commande TELL exige que le correspondant soit en session).
2. la note constitue l'information que vous transmettez habituellement par lettre. C'est donc souvent plus long qu'un message. Une note n'exige donc pas que le correspondant soit en session ce qui implique que le système va la mémoriser jusqu'à ce que le destinataire soit prêt à la lire.

### 5.3.1 Echange de messages entre machines virtuelles: TELL

La commande TELL a la forme suivante:

```
TELL nom [AT noeud] message
```

Figure 73. Commande EXEC TELL

Remarque: nom peut être un userid (si le destinataire travaille sur le même ordinateur que vous) ou un surnom défini dans un fichier de noms (Voir «Le fichier de noms.»)

Si le destinataire n'est pas en session au moment de l'envoi du message vous en serez immédiatement averti.

Si le destinataire est en mode fullscreen à ce moment, il recevra le message dès son prochain ENTER (ou touche TRANSM).

"noeud" doit être indiqué si "nom" n'est pas repris dans un NAMES FILE et que le destinataire est connecté à un autre ordinateur:

```
"noeud"
vaudra BLIULG11 si le destinataire est sur le superordinateur.
```

### 5.3.2 Refuser les messages.

Pour éviter d'être dérangé par la réception de messages on peut toujours entrer la commande CP SET MSG OFF. Plus tard, si l'on désire à nouveau réceptionner les messages on fera SET MSG ON.

L'émetteur d'un message est averti si le destinataire a refusé la réception du message.

### 5.3.3 Le fichier de noms.

Pour communiquer avec quelqu'un vous devez connaître son userid et l'identificateur de l'ordinateur sur lequel il travaille (excepté s'il travaille sur le même ordinateur que vous). En téléphonie nous dirions qu'il faut connaître son numéro de téléphone et le numéro de sa zone. Ainsi au SEGI, le superordinateur a l'identificateur BLIULG11. Ces identificateurs sont appelés nodes.

Il serait laborieux de mémoriser le userid et le node de tous ses interlocuteurs potentiels. Pour cette raison il est possible d'en garder une liste dans un fichier de noms appelé NAMES FILE. On peut également compléter le NAMES FILE avec d'autres informations telles que l'adresse ou le numéro de téléphone du correspondant. Ce fichier permet ainsi d'associer un surnom à chaque destinataire, surnom qui sera utilisé dans les dialogues en lieu et place du userid et du node.

La création du NAMES FILE se fait au moyen de la commande

LNAME

Figure 74. Commande EXEC LNAME

LNAME affiche un panneau fullscreen dans lequel vous devez remplir les zones suivantes:

**nick** pour indiquer le surnom utilisé  
**userid** le userid du destinataire  
**node** le node du destinataire. Omis s'il s'agit du même ordinateur que l'émetteur.

**Vous devez vous définir vous même en premier lieu.**

Afin que les notes contiennent des informations plus précises on peut compléter les zones suivantes également

**name** le nom du destinataire  
**phone** son numéro de téléphone  
**address** son adresse

Normalement, une copie de toutes les notes envoyées et reçues est sauvée automatiquement dans le fichier intitulé ALL NOTEBOOK. Pour ranger séparément la correspondance relative à une personne on peut lui affecter un fichier particulier dont le filename sera renseigné dans la zone NOTEBOOK. On peut définir aussi une liste de distribution afin de pouvoir envoyer une note ou un message à un groupe de personnes en une seule opération. Pour cela il faut indiquer le nom de la liste dans la zone nick et le surnom ou le userid des destinataires dans la zone "List of Names".

Après chaque définition d'une nouvelle personne ou d'une nouvelle liste on fait PF2 pour l'enregistrer. On quitte la commande LNAME en faisant PF3.

### 5.3.4 Transmission de notes.

Le développement du courrier électronique, avec des machines et des réseaux couvrant le monde entier, a donné lieu au remplacement des commandes standards du CMS (commande NOTE principalement) par des commandes plus puissantes permettant d'adresser des correspondants sur les réseaux les plus divers et de gérer son courrier de manière plus performante et plus conviviale. On se référera au chapitre «Le courrier électronique» à la page 80 pour une introduction à ces commandes (POST et MAILBOOK principalement).

### 5.3.5 Transmission de fichiers.

Il est possible d'envoyer des fichiers au moyen de la commande

```
{ SENDFILE } fn ft fm TO surnom [AT noeud]
  SF
```

Figure 75. Commande EXEC SENDFILE

"noeud" doit être indiqué si "surnom" n'est pas repris dans un NAMES FILE et que le destinataire est connecté à un autre ordinateur.

La réception des fichiers sur minidisque s'effectue grâce aux commandes RDRLIST et RECEIVE.

## 5.4 Les interpréteurs de commandes.

### 5.4.1 Introduction.

Un interpréteur de commandes permet à l'utilisateur:

- de créer des procédures de commandes, c'est-à-dire de nouvelles commandes CP ou CMS;
- de créer de nouvelles sous-commandes de l'éditeur XEDIT;
- de se définir un "profile" général, c'est-à-dire un ensemble de commandes CP ou CMS qui seront automatiquement exécutées à chaque entrée en session au moment du LOGON (voir «Définir son PROFILE.» à la page 36);
- de se définir un "profile" particulier à l'éditeur, c'est-à-dire un ensemble de sous-commandes qui pourront être exécutées automatiquement lors de l'appel de l'éditeur XEDIT;
- et d'une manière générale, de programmer des applications diverses: calcul, gestion de fichiers, ...

**Remarque :** on utilise souvent le terme procédure de commandes pour désigner un programme écrit pour un interpréteur de commandes.

Nous disposons actuellement de trois interpréteurs de commandes:

- EXEC, l'interpréteur original du CMS;
- EXEC2, une version plus récente de cet interpréteur;
- SPI ("System Product Interpreter"), un interpréteur pour le langage REXX ("Restructured eXtended eXecutor Language") disponible en VM à partir du VM/SP3.

Compte tenu des performances du SPI, et des caractéristiques du langage REXX (relatives notamment à la programmation structurée), c'est ce dernier langage, présenté ci-dessous (voir «Le langage REXX.» à la page 39), qui devrait être utilisé pour de nouvelles applications. Les langages EXEC et EXEC2 ne sont pas repris dans cette brochure.

Les programmes écrits dans ces différents langages résident dans des fichiers de type EXEC ou XEDIT selon qu'il s'agit de procédures de commandes ou de sous-commandes de l'éditeur.

La distinction entre les différents langages se fait au moyen de la première instruction d'un programme:

**en REXX**, la première instruction doit être un commentaire délimité par /\* et \*/.  
**en EXEC2**, la première instruction doit être &TRACE.

Si la première instruction d'un programme n'est ni un commentaire REXX ni &TRACE, le programme sera considéré comme un programme EXEC.

Certaines considérations communes aux trois langages sont présentées sommairement ci-dessous; elles concernent:

- la façon d'appeler un interpréteur de commandes;
- quelques commandes CMS particulièrement utiles dans des procédures de commandes;
- l'échange de données par l'intermédiaire du "console stack"<sup>9</sup>.

#### 5.4.1.1 Appel d'un interpréteur de commandes.

Un interpréteur de commandes est appelé comme suit:

```
[EXEC] filename [arg1 arg2 ...]
```

Figure 76. Commande CMS EXEC

Remarque: filename indique le nom du fichier dans lequel réside le programme, et arg1, arg2... les arguments à transmettre.

#### 5.4.1.2 Commandes CMS utiles dans les procédures de commandes.

Pour des détails, Voir le "CMS Command Reference" :

**SET ABBREV ON/OFF** accepter(ON) ou refuser(OFF) les abréviations. (Défaut : ON).

**SET CMSTYPE HT/RT** supprimer(HT) ou rétablir(RT) l'envoi de messages au terminal par certaines commandes CMS.

**SET EXEC TRAC ON/OFF** établir(ON) ou supprimer (OFF) le "tracing" d'un programme; le tracing permet de contrôler le déroulement de l'exécution d'un programme. (Défaut: OFF).

#### EXECIO

cette commande réalise des échanges de données entre une procédure de commandes et son environnement. Elle permet

- de lire des données sur disque ou dans le lecteur virtuel et de les mémoriser dans une variable ou dans le "program stack" de manière à les rendre accessible à une procédure de commandes;
- d'écrire des données, en provenance d'une variable ou du "program stack", sur disque ou sur l'imprimante ou le "punch" virtuels
- d'exécuter des commandes CP et de mémoriser les informations qui en résultent dans le "program stack";

#### GLOBALV

permet de définir et de manipuler des variables globales, c'est-à-dire des variables qui pourront être partagées entre plusieurs procédures de commandes.

#### MAKEBUF

créé un nouveau buffer dans le "program stack".

#### DROPBUF

supprime le buffer le plus récent du "program stack".

#### SADT

empile des informations décrivant l'état d'un minidisque dans le "program stack" (taper HELP SEGI SADT pour voir le mode d'emploi à l'écran);

#### GETFMADR

empile dans le "program stack" le premier filemode et la première adresse virtuelle disponibles pour connecter et accéder un minidisque (taper HELP SEGI GETFMADR);

#### VMFCLEAR

efface l'écran.

#### EXTRACT

EXTRACT est une sous-commande de XEDIT; elle permet à une procédure de sous-commandes de l'éditeur d'obtenir des informations en provenance des variables internes de l'éditeur (voir Editor Command and Macro Reference).

### 5.4.2 Le "console stack".

Les différents interpréteurs de commandes permettent de procéder à des échanges de données par l'intermédiaire du "console stack". Celui-ci est représenté à la Figure 77 à la page 39 ; il a la structure d'une pile<sup>10</sup> de "buffers" (zones de mémoire virtuelle). Il comprend le "terminal input buffer" (appelé aussi "system external event queue") et le "program stack" (appelé aussi "system-provided data queue"):

le "**terminal input buffer**" est une file d'attente ("queue")<sup>11</sup> qui mémorise toute ligne entrée au terminal dans la "CMS command line" pendant qu'une commande précédente est en train de s'exécuter. La commande précédente en question peut

<sup>9</sup> Le "console stack" est un système de "buffers" (zones de mémoire virtuelle) par lequel une procédure de commandes communique avec son environnement (console, fichiers, ...).

<sup>10</sup> Une pile mémorise des données qui pourront être récupérées dans l'ordre inverse de celui où elles ont été entrées; c'est une structure de données de type LIFO ("Last In First Out").

<sup>11</sup> Une file d'attente mémorise des données qui pourront être récupérées dans l'ordre où elles ont été entrées; c'est une structure de données de type FIFO ("First In First Out").

être une procédure de commandes; dans ce cas, elle peut très bien lire les lignes mémorisées dans le "terminal input buffer" pour en faire un traitement particulier. Le "terminal input buffer" est donc un moyen de communications entre une procédure de commandes et la console de l'utilisateur;

le "**program stack**" est une pile de buffers par l'intermédiaire desquels peuvent se faire les échanges de données entre une procédure de commandes et son environnement : lecteur, imprimante, "punch" virtuels et fichiers (voir commande EXECIO ci-dessus).

Initialement, le "program stack" comprend un seul buffer (le buffer 1); l'utilisateur peut en créer de nouveaux ou en supprimer avec les commandes CMS MAKEBUF ou DROPBUF respectivement. C'est toujours le dernier buffer créé qui est accessible pour y mémoriser des données; ce buffer peut être utilisé à la fois comme une pile ou comme une file d'attente, selon la façon dont les commandes appropriées y mémorisent les données.

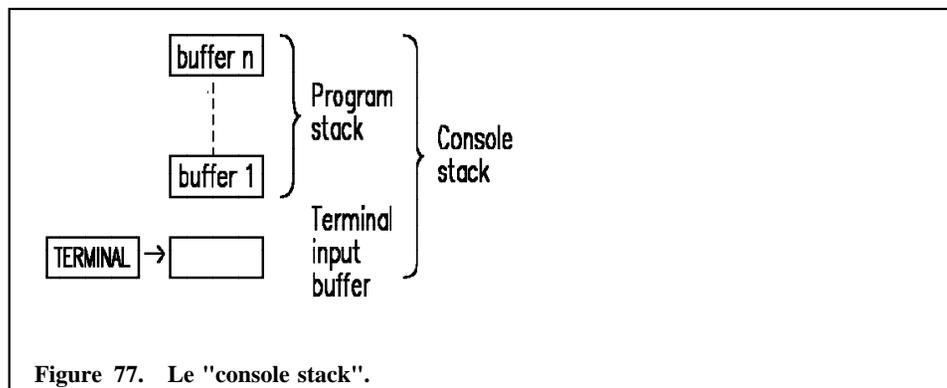


Figure 77. Le "console stack".

Quand une commande doit lire des données,

- elle les cherche d'abord dans le "program stack";
- si ce dernier est vide, elle les cherche dans le "terminal input buffer";
- si ce dernier est vide, elle les attend au terminal qui passe en mode VM READ.

Quand un programme se termine, toutes les lignes qui subsistent dans le "console stack", qui n'ont donc pas été lues par le programme, sont traitées comme des commandes CMS.

**Remarque :** pour éviter de modifier l'environnement, il est recommandé que tout programme qui utilise le "console stack" se définisse son propre buffer dans le "program stack" (commande MAKEBUF) et le libère en fin d'exécution (commande DROPBUF).

### 5.4.3 Le langage REXX.

**Remarque :** la commande HELP REXX MENU affiche un menu à partir duquel on peut obtenir la description de toutes les instructions et fonctions internes du REXX.

Un programme REXX est composé d'une série de clauses dont la première ne comprend qu'un commentaire. Chaque clause est composée d'éléments appelés "tokens" séparés par des blancs ou autres délimiteurs et terminée par point-virgule. Si le point-virgule n'est pas explicitement spécifié, il est impliqué par la fin de ligne, sauf si la ligne se termine par virgule. La virgule, utilisée en fin de ligne permet de prolonger une clause sur plusieurs lignes.

Les différents éléments (tokens) d'une clause peuvent être:

- des commentaires** exemple: /\* commentaire \*/. Les commentaires agissent comme des délimiteurs;
- des strings** exemples: 'string1', "string2", "l'école", 'l'école', " (ce dernier exemple représente le string nul);
- des nombres** exemples: 123, -2.42, 0.123E4; les nombres sont traités comme des strings; ainsi, les exemples précédents pourraient aussi s'écrire '123', "-2.42", ... (précision par défaut: 9 chiffres; l'instruction NUMERIC DIGITS permet de modifier la précision);
- des opérateurs** (cf. Figure 78 à la page 40); ils agissent comme des délimiteurs;
- des caractères spéciaux** parenthèses, virgule, point-virgule et deux points; ils agissent comme des délimiteurs;
- des symboles** groupes de caractères tirés de l'alphabet (A-Z, a-z), des chiffres décimaux (0-9) et des caractères @, #, \$, ç, ., ], ? et \_.

Un symbole peut représenter:

- **un mot-clé du langage**,
- **un nombre** (à noter que 2.1 est un symbole mais -2.1 ne l'est pas puisque le signe - n'est pas repris dans la liste des caractères ci-dessus); tout symbole dont le nom commence par un chiffre ou un point est un symbole constant;
- **un nom de variable** (initiale autre que chiffre et point);
- **une étiquette** (tout symbole suivi de deux points).

Tout symbole a une valeur qui, par défaut, est le string composé des caractères de son nom éventuellement transcodés en majuscules (exemple: la valeur initiale du symbole Ecole est ECOLE);

#### 5.4.3.1 Variable et clause d'assignation.

Un symbole dont le nom ne commence ni par un point ni par un chiffre peut être utilisé comme variable. La clause d'assignation permet d'attribuer explicitement une valeur à un symbole:

```
variable=expression;
Exemple: total=3+2;
```

Le nom d'une variable peut être un symbole composé. Exemple:

```
table.ligne.colonne
```

Les différents éléments d'un symbole composé sont séparés par des points. Le premier élément, le radical ("stem"), doit avoir la syntaxe d'un nom de variable. Les éléments suivants peuvent être des variables ou des symboles constants. Les symboles composés permettent de construire des tableaux. Exemple:

```
table.1.1='Dupont'; table.1.2=2340;
table.2.1='Durand'; table.2.2=3720;
table.3.1='Georges'; table.3.2=1200;
```

Après cela, si ligne=3 et colonne=2, alors table.ligne.colonne a la valeur 1200.

Remarque: l'assignation

```
table.=0;
```

attribue la valeur 0 à toutes les variables dont les noms peuvent être dérivés du radical table.

### 5.4.3.2 Opérateurs, fonctions internes et expressions.

Une expression comprend des termes séparés par des opérateurs et/ou des parenthèses. Un terme peut être un string, un symbole ou un appel de fonction. REXX est pourvu d'un ensemble très riche de fonctions internes ("built-in functions", voir le System Product Interpreter Reference); de plus, l'utilisateur peut lui-même écrire des fonctions REXX. Les principaux opérateurs, avec leurs niveaux de priorité sont présentés dans la figure ci-après. Les règles d'évaluation des expressions sont analogues à celles des langages FORTRAN et PL/1.

Priorité	Opérateurs	
1	+ - ^	préfixes: signes + - et négation
2	**	exponentiation
3	* /	multiplication et division
	% //	quotient entier et modulo
4	+ -	addition et soustraction
5	blanc	concaténation avec/sans blanc entre les opérandes
6	= == <>	comparaisons
	^= ^=	
	>= <=	
7	...	
	&	ET logique
8	&&	OU et OU EXCLUSIF logiques

**Figure 78. PRIORITE DES OPERATEURS REXX**

**Remarque :** en REXX, les données n'ont pas de type ("typeless"); toute donnée est mémorisée comme un string. C'est ce qui explique l'existence d'un opérateur d'identité ==. Exemple:

```
si x=123 et y=0123,
alors x=y est vrai
mais x==y est faux (x n'est pas identique à y).
```

### 5.4.3.3 Commandes.

Toute clause qui n'est pas reconnue comme un commentaire, un label, une assignation ou une instruction (une instruction commence par un mot-clé, voir « Instructions.» à la page 41) est considérée comme une commande.

En général, une commande est représentée par une expression; SPI évalue cette expression et passe la commande qui en résulte à l'environnement; celui-ci exécute la commande puis rend le contrôle au programme REXX en lui transmettant un "return code" par l'intermédiaire de la variable spéciale RC.

Par défaut, l'environnement est CMS pour une procédure de commandes (fichier de type EXEC) et XEDIT pour une procédure de sous-commandes de l'éditeur XEDIT (fichier de type XEDIT). Une instruction REXX, ADDRESS, permet de définir explicitement l'environnement.

Quand l'environnement est CMS, le système vérifie en premier lieu s'il s'agit d'une procédure EXEC et en second lieu s'il s'agit d'une commande CMS. Si ce n'est ni l'une ni l'autre, la commande est transcodée en majuscules et passée au CP.

**Remarque :** on peut toujours invoquer le CP explicitement en préfixant la commande par CP.

Il est parfois nécessaire d'utiliser des apostrophes ou guillemets pour éviter que certaines parties d'une commande soient mal interprétées par SPI. Exemple :

```
ERASE * LISTING; /* erreur : * est interprété comme
                  l'opérateur de multiplication*/
ERASE "*" LISTING /* ok, sauf si ERASE ou LISTING sont
                  utilisés ailleurs comme variables
                  et ont des valeurs différentes de
                  ERASE et LISTING respectivement */
"ERASE * LISTING" /* ok */
```

#### 5.4.3.4 Instructions.

Une instruction comprend une ou plusieurs clauses dont la première commence par un mot-clé qui identifie l'instruction.

#### Instructions de transfert de données.

**SAY [expression]** affichage au terminal de la valeur de l'expression.

**Remarque :** la fonction interne FORMAT permet de faire de la mise en page.

**[PARSE [UPPER]] PULL [template]** lecture d'une ligne de données en provenance du "console stack". PARSE doit être spécifié si UPPER l'est. UPPER transcode les données lues en majuscules; c'est l'option par défaut si ni PARSE ni UPPER n'est spécifié. A titre exemplatif, on supposera que la ligne de données qui est lue est la suivante:

```
donnee1 donnee2 (option1 option2
```

Le "template" (gabarit) peut être une liste de variables. Exemple:

```
PULL x1 x2 x3
```

Les éléments de données sont séparés par des blancs et affectés aux variables dans l'ordre: "donnee1" à x1, "donnee2" à x2 et le reste, "(option1 option2" à x3. Par contre,

```
PULL x1 x2 x3 x4 x5
```

affecterait "donnee1" à x1, "donnee2" à x2, "(option1" à x3 et "option2" à x4; la variable x5 aurait la valeur string nul.

Le template peut contenir des points; un point tient la place d'une variable, il est associé à une donnée mais celle-ci n'est pas affectée. Exemple:

```
PULL x1 . . x2
```

La valeur "donnee1" sera affectée à x1 et "option2" à x2.

Le "template" peut aussi contenir des "patterns", strings ou nombres entiers. Exemple :

```
PULL x1 "( " x2 x3
```

Dans ce cas, on peut considérer que le template est séparé en deux sections : celle qui précède le pattern (x1) et celle qui le suit(x2 x3). Il en est de même pour les données qui, dans notre exemple, sont séparées en:

```
première section: donnee1 donnee2
deuxième section: option1 option2
```

Les différentes sections du template et des données sont respectivement associées et x1 reçoit la valeur "donnee1 donnee2", x2 reçoit la valeur "option1", x3 reçoit la valeur "option2".

Si les patterns sont des nombres entiers, ils désignent des numéros de colonnes sur la base desquelles la ligne de données sera séparée en différentes sections.

Les patterns peuvent être représentés par des noms de variables entre parenthèses.

**[PARSE [UPPER]] ARG [template]** permet d'affecter aux variables du template les valeurs des arguments spécifiés lors de l'appel du programme (sous-programme, fonction ou procédure). La fonction interne ARG permet entre autre chose d'obtenir le nombre de ces arguments. Exemple: n=ARG();

**Remarque :** l'instruction PARSE permet encore de réaliser d'autres opérations (voir le System Product Interpreter Reference).,

**PUSH [expression]** place la valeur de l'expression au sommet de la pile que constitue le dernier buffer du "program stack".

**Remarque :** la fonction interne QUEUED() permet d'obtenir le nombre de ligne mémorisées dans le "console stack". Exemple :

```
n=QUEUED();
```

**QUEUE [expression]** place la valeur de l'expression au bout de la file d'attente que constitue le dernier buffer du "program stack".

**Instructions de contrôle** Elles permettent de définir la structure du programme.

- Groupe de clauses:

```
DO;
clause1;
clause2;
...
END;
```

- Alternative: l'expression indiquée ne peut avoir que l'une des valeurs 0 (faux) ou 1 (vrai):

```
IF expression THEN clause1;
[ELSE clause2;]
```

**Remarque :** aucune branche de l'alternative ne peut être vide. On ne peut donc écrire:

```
IF expression THEN; /*incorrect*/
ELSE clause2;
mais bien
IF expression THEN NOP; /*correct*/
ELSE clause2;
```

NOP est l'instruction nulle ("No OPeration").

- Sélection: les expressions indiquées ne peuvent avoir que les valeurs 0 ou 1:

```
SELECT;
WHEN expression1 THEN clause1;
WHEN expression2 THEN clause2;
...
[OTHERWISE clause;]
END;
```

(Même remarque que ci-dessus à propos de l'instruction NOP).

- Répétitions: Principales formes:

```
DO expression; /*Répétition de (expression) fois*/
clauses;
END;
```

```
DO var=exp1 TO exp2 [BY exp3] [FOR exp4];
clauses; /*exp4 indique le nombre maximum*/
END [var]; /*de répétitions*/
```

Dans les deux exemples suivants, les expressions ne peuvent avoir que les valeurs 0 ou 1.

```
DO WHILE exp; /*L'expression est évaluée*/
clauses; /*et testée en début de boucle*/
END;
```

```
DO UNTIL exp; /*L'expression est évaluée*/
clauses; /*et testée en fin de boucle*/
END;
```

**Remarque :** l'instruction LEAVE permet de quitter une boucle avant que la condition de fin de boucle ne soit atteinte. L'instruction EXIT permet de quitter un programme.

### Instructions de reprise d'erreur et de "debugging".

**SIGNAL ON ERROR** cette instruction provoque une rupture de séquence et renvoie à la clause précédée du label ERROR: dès qu'une commande donne un "return code" différent de 0. La variable spéciale RC contient la valeur du "return code". La variable spéciale SIGL contient le numéro de ligne de la dernière clause exécutée.

**SIGNAL ON HALT** cette instruction provoque une rupture de séquence et renvoie à la clause précédée du label HALT: dès que la commande CMS HI ("Halt Interpretation") a été exécutée.

**TRACE options** cette instruction permet de contrôler l'exécution du programme. Exemple: TRACE ALL provoque l'affichage au terminal de toutes les clauses qui sont exécutées. Pour les autres options, voir System Product Interpreter. L'instruction TRACE permet notamment de faire du "debugging" interactif.

**Remarque :** rappelons que le "tracing" peut aussi être établi ou arrêté par l'instruction SET EXEC TRAC. De plus, les commandes CMS immédiates TS ("Trace Start") et TE ("Trace End") permettent respectivement de déclencher et d'arrêter le tracing de façon asynchrone, c'est-à-dire pendant que le programme s'exécute.

**Autres instructions** REXX possède encore un certain nombre d'autres instructions. Signalons:

- la possibilité de définir des routines (fonctions, sous-programmes, procédures), routines qui peuvent être récursives et qui permettent de mettre en oeuvre les concepts de variables locales et globales.
- l'instruction INTERPRET qui permet d'exécuter des instructions construites dynamiquement.

### 5.4.3.5 Exemples.

**Le fichier PROFILE:** la première commande introduite après l'IPL CMS provoque au préalable l'exécution de la procédure EXEC figurant dans un fichier PROFILE EXEC accessible à la machine de l'utilisateur. Ce fichier est particulièrement utile à l'utilisateur qui peut ainsi définir son environnement de travail, par exemple:

- en attribuant aux PF Keys d'un terminal 3270 les fonctions désirées,
- en définissant les bibliothèques TEXT ou MACRO employées tout au long de la session,
- en affectant à son punch ou printer des caractéristiques particulières,
- en accédant à des minidisques qui ne le sont pas en standard; etc...

Pour empêcher l'exécution du PROFILE EXEC, il faut entrer, comme première commande après l'IPL CMS, la commande ACCESS (NOPROF).

```

/* Exemple de fichier PROFILE EXEC */
CP SET EMSG ON
CP LINK UZZZZ01 191 199 RR
ACCESS 199 Z
"CP SET PF01 IMMED LISTFILE * *"
CP SET PF06 RETRIEVE
FILEDEF INMOVE R
FILEDEF OUTMOVE T
CP SET PF02 IMMED MOVE

```

**Utilisation du "program stack":** la procédure suivante définit un mini-disque temporaire à une adresse et avec un filemode disponibles. L'appel de la procédure se fait comme suit :

```
[EXEC] TEMPO [cyl]
```

L'argument cyl est le nombre de cylindres. S'il n'est pas spécifié, il sera demandé par le programme à l'utilisateur.

```

/*          TEMPO          */
/*      Définition d'un disque temporaire      */
/*  à une adresse et avec un file mode disponibles  */
signal on error;
clrscrn;          /*Effacement de l'écran*/
if arg(>)>0 then arg cylindres .;
/*Réception du premier argument s'il a été spécifié*/
do while verify(cylindres,'0123456789')<>0;
/*Vérification que cylindres
ne contient que des chiffres*/
say "Entrer le nombre de cylindres";
pull cylindres .;
end;
getfmadr; /*obtention d'un file mode et d'une adresse
disponibles*/
pull . mode adresse;
define t3380 as adresse cyl cylindres;
push tempo ;          /*label du disque*/
push yes ;          /*réponse pour la commande format*/
set cmstype ht;          /*suppression des messages de
la commande format*/

format adresse mode;
set cmstype rt;
say "Disque temporaire : " mode "/" adresse
exit;
error: say "erreur à la ligne" sigl ,
"- return code =" rc;

```

### 5.4.3.6 CMS Pipelines

Le logiciel CMS Pipelines peut être considéré comme une extension du REXX. Il comporte notamment une commande PIPE qui permet de définir des pipelines.

Un pipeline est une commande composée; il comprend plusieurs commandes qui s'exécutent séquentiellement, l'output de l'une étant l'input de la suivante. Le CMS Pipelines permet d'écrire des commandes ou des procédures de commandes dans un stype comparable à celui de UNIX.

Exemple :

```
PIPE CP QUERY NAMES | SPLIT , | SORT | CONSOLE
```

L'output de la commande "CP QUERY NAMES", qui a la forme suivante :

```

ANTOINE -H04LC62 , IBMCE -H04LC6C , OP1 -T04300V0
UZ15111 -T04300VB, UZ24523 -H0DL304 , UZ24519 -H0DL307
WBALLENS-T0112241, UZ24520 -T0D1019V, UFERTONS-T01Z1V9
UZ131A3 -H04E3A2 , WDOHET -T0111161, STMARD -H04LC67
DEMOITIE-T01K1V8 , U2150SM -T01I1V13, UZ24512 -H0DL303
GUIDEVM -T012P8V2, U305001 -T01B1V2 , U2141JV -T01H1V11
U2152B5 -T01I1V11, ADMETUD2-T01K1V5 , U016404 -H04ECDF

```

est passé comme input à la commande SPLIT (qui, dans le jargon CMS Pipeline est un **filtre**); "SPLIT ," crée un nouveau record chaque fois qu'une virgule est rencontrée. L'output de SPLIT est passé à SORT, filtre qui trie les records dans l'ordre alphabétique. L'output de SORT est passé à CONSOLE, un des **devices** du CMS Pipelines, et affiché à l'écran dans la forme suivante :

```

ANTOINE -H04LC62
ADMETUD2-T01K1V5
DEMOITIE-T01K1V8
GUIDEVM -T012P8V2
IBMCE -H04LC6C
OP1 -T04300V0
STMARD -H04LC67
U016404 - H04ECDF
...

```

Pour plus de détails, voir «Bibliographie» à la page 113.

## 5.5 La commande HELP.

La commande CMS HELP permet d'obtenir plusieurs menus au terminal, dont les principaux sont repris ci-dessous:

1. Le premier menu, est un menu IBM, que l'on peut aussi obtenir en tapant

```
HELP MENU [(options)]
```

**Figure 79. Commande HELP MENU**

Ce menu donne la liste des menus pour lesquels on peut obtenir des informations au terminal en tapant

```
HELP composant MENU [(options)]
```

**Figure 80. Commande HELP composant**

Les composants possibles sont:

AMMOTHER	COMMANDS	DUMP	INCH	PVM	SFSQUERY	VMDS
AMMR	CP	DUMPSCAN	INDICATE	QUERY	SPTAPE	VMFINS
APPCVM	CPOTHER	DUMPCVIEW	IUCV	RDEVICE	SQLDS	VMFSIM
AVS	CPQUERY	EDIT	LANRES	REXX	SRPI	VMSES
BORDER	CPSET	EXEC	MACRO	ROUTINE	STORE	VSCREEN
CMS	CPUTIL	EXEC2	MACROS	ROUTINES	TERMINAL	WINDOW
CMSQUERY	CRR	FILESERV	MONITOR	RSCS	TRACE	XEDIT
CMSSET	DEFINE	FUNCTION	PIPE	SET	TRSOURCE	XLINK
CMSUTIL	DETACH	GROUP	PREFIX	SFSADMIN	TSAF	XSPool
COMMAND	DISPLAY	HMF	PURGE			

2. Le deuxième menu, est un menu IBM, que l'on peut aussi obtenir en tapant

```
HELP COMMANDS MENU [(options)]
```

**Figure 81. Commande HELP COMMANDS MENU**

Ce menu donne la liste des commandes IBM (CP et CMS uniquement) pour lesquelles on peut obtenir des informations au terminal en tapant

```
HELP [composant] commande [(options)]
```

**Figure 82. Commande HELP commande**

CMS est le composant par défaut.

3. Le troisième menu, est un menu IBM, que l'on peut aussi obtenir en tapant

```
HELP SEGI [TASK] [(options[])]
```

**Figure 83. Commande HELP SEGI**

Ce menu donne accès

- aux grandes **catégories** de logiciels mis à la disposition des utilisateurs sur l'ensemble des systèmes du SEGI (LANGAGES, PACKAGES, NETWORKS, SPACEMAN, UTILITAI) et dont on peut obtenir une liste plus détaillée en tapant:

```
HELP catégorie [TASK] [(options[])]
```

**Figure 84. Commande HELP catégorie TASKS**

- aux listes des commandes et procédures directement disponibles sur chaque **système** d'exploitation utilisé au SEGI (SEGIVMx) et que l'on peut obtenir directement en tapant:

```
HELP système [MENU] [(options[])]
```

**Figure 85. Commande HELP système MENU**

### Remarques:

1. Les informations plus complètes sur les commandes et ensembles de programmes (**composant**) qui apparaissent dans les menus précédents peuvent s'obtenir au terminal en tapant:

```
HELP composant [TASK] [(options [])]
```

ou

```
HELP composant commande ou programme [(options [])]
```

ou

```
HELP SEGI commande [(options [])]
```

**Figure 86. Commande HELP**

Les principaux **composants** sont :

<b>BMDP</b>	programmes de statistique.
<b>CLASSES</b>	classes de travaux "batch" utilisables sur différents systèmes
<b>DIRMAINT</b>	commandes relatives à la gestion d'une machine virtuelle;
<b>EXECSERV</b>	commandes de type utilitaire;
<b>MICROS</b>	liste de produits "micros" disponibles sur le superordinateur.
<b>PUBLIB</b>	programmes de type utilitaires;
<b>SPACE</b>	commandes de gestion des mini-disques;
<b>STATPROG</b>	programmes de statistique;
<b>STATSP</b>	sous-programmes de statistique;

2. La commande HELP permet aussi d'obtenir au terminal la signification d'un message d'erreur en tapant

HELP message

Figure 87. Commande HELP message

**Remarque:** message désigne l'identification d'un message d'erreur amputée des caractères de 4 à 6. Exemple: pour avoir l'explication du message

```
DMSEIO622E INSUFFICIENT FREE STORAGE FOR EXECIO RC=41
```

il faut taper

```
HELP DMS622E
```

**Remarque :** pour recevoir les numéros des messages d'erreur au terminal, il faut avoir défini EMSG ON par la commande SET.

3. Les options de la commande HELP peuvent être ALL, DESC, FORM, PARM, OPT, NOTE ou ERR selon que l'on désire la totalité du texte (ALL, option par défaut), la partie description (DESC), la partie syntaxe (FORM), la description des paramètres (PARM), les options (OPT), les remarques et exemples (NOTE) ou les messages d'erreurs (ERR).
4. La commande HELP HELP permet d'obtenir un complément d'informations sur l'exploitation globale des panneaux.
5. Plutôt que de taper l'entièreté des commandes décrites ci-dessus, on peut utiliser la commande HELP par étapes en naviguant dans les différents menus proposés.

## 5.6 Modes de fonctionnement et procédures d'interruption

### 5.6.1 Terminal de type 3270

En fin de dernière ligne de l'écran sont indiqués le nom de l'ordinateur (BLIULG11 pour le superordinateur) sur lequel vous êtes en session et l'état dans lequel se trouve le terminal.

Cet état peut être:

**CPREAD** le CP est en attente d'input au clavier; on ne peut entrer dans cet état que des commandes CP. Il faut également être dans cet état pour entrer la commande LOGON d'où la nécessité d'entrer une ligne nulle lorsque l'écran contient l'indication de connexion au VM sous la forme d'une mire.

**VMREAD** la machine virtuelle (CMS) a lancé une opération de lecture au terminal et attend la réponse.

**RUNNING MORE** la machine virtuelle n'est pas en attente d'input au terminal. signifie qu'une opération d'affichage sur l'écran a rempli celui-ci. On peut provoquer le passage à la page suivante en appuyant sur la touche CLEAR. Si on ne fait rien, le saut de page a lieu au bout d'une minute.

**HOLDING** on obtient cet état en appuyant sur la touche ENTER dans l'état MORE, afin d'éviter le saut de page automatique au bout d'une minute.

D'autres états sont possibles, mais non décrits ici: XEDIT, DEBUG, ...

### 5.6.2 Terminal "ligne"

Sur un terminal "ligne" (TTY, ...), l'état dans lequel on se trouve n'est pas indiqué en permanence. Deux états sont signalés: mode CP et mode CMS qui sont indiqués notamment lorsque l'on passe de l'un à l'autre. Les états MORE et HOLDING n'existent pas. La distinction VMREAD, RUNNING n'est pas affichée.

### 5.6.3 Passage d'un état à l'autre et interruption

On peut passer à l'environnement CP en entrant depuis un autre environnement (CMS) la commande #CP ou en appuyant sur la touche PA1 (terminal 3270). On revient à l'environnement initial par la commande CP BEGIN.

On provoque une interruption de la machine virtuelle (c'est-à-dire de la commande CMS en cours d'exécution) en envoyant une ligne nulle (n'importe quel caractère ou le break sur un TTY). On se retrouve à ce moment dans l'état VMREAD. Si on entre une ligne nulle, l'exécution reprend au point d'interruption; si on entre HX, l'exécution est terminée; si on entre HT, l'exécution se poursuit mais l'affichage ne se fait plus (il reprend en entrant la commande RT); si on entre un autre texte, il est enregistré et pris comme input lorsque la commande se termine.

## 5.7 Programmation sous CMS

### 5.7.1 La compilation

Il existe une commande particulière pour appeler chacun des compilateurs disponibles au SEGI (cf. «Appel des compilateurs et interpréteurs» à la page 48). Dans le présent paragraphe, nous présentons les caractéristiques communes à chacun des compilateurs.

Si le compilateur (ou l'assembleur) a besoin de librairie(s) de macro-instructions (MACLIB) il convient de les rendre accessibles au moyen de la commande suivante avant d'appeler le compilateur :

GLocal MACLIB fn1 [fn2 ...]

Figure 88. Commande CMS GLOBAL MACLIB

L'effet d'une commande GLOBAL MACLIB demeure jusqu'au prochain IPL CMS ou jusqu'à la prochaine commande GLOBAL MACLIB.

Le compilateur crée généralement 2 fichiers sur le disque A:

1. Un fichier fn TEXT A1 qui contient le module objet
2. Un fichier fn LISTING A1 qui contient le listing produit par le compilateur

Les messages d'erreur sont envoyés dans le fichier listing mais également au terminal.

Les compilateurs VS FORTRAN, VS PASCAL, VS COBOL, PL/1 et C créent des modules objets compatibles avec l'architecture 370 aussi bien que XA. Ces modules objets ont un **mode d'adressage** qui leur permet d'utiliser des adresses de 24 ou 31 bits et un **mode de résidence** qui, lors du chargement, leur permet de profiter d'une mémoire virtuelle qui peut aller jusqu'à 999Mb. On dit que leur AMODE (Mode d'Adressage) et leur RMODE (Mode de Résidence) sont ANY. Ils pourront donc être exploités dans une machine virtuelle d'architecture 370 dont la mémoire est limitée à 16Mb aussi bien que dans une machine virtuelle d'architecture XA qui peut avoir jusqu'à 999Mb de mémoire.

En ASSEMBLER H, le mode d'adressage et le mode de résidence peuvent être définis explicitement dans un programme au moyen des instructions AMODE et RMODE (cf. «Bibliographie» à la page 113).

Les modules objets produits par les compilateurs (fichiers de type TEXT) peuvent être regroupés dans des bibliothèques de modules objets (fichiers de type TXTLIB) au moyen de la commande CMS TXTLIB.

## 5.7.2 Le chargement

La commande LOAD permet de charger un module objet en mémoire virtuelle et de résoudre les références externes contenues dans le programme en parcourant les fichiers dont le ft est TEXT et le fn identique à la référence externe à résoudre. Ensuite, si toutes les références ne sont pas résolues, CMS parcourt les bibliothèques de sous-routines (TXTLIB) qui auront, au préalable, été définies par la commande GLOBAL TXTLIB.

GLocal TXTLIB fn1 [fn2][fn3 ...]

Figure 89. Commande CMS GLOBAL TXTLIB

Les bibliothèques utiles pour chaque compilateur sont indiquées au paragraphe «Appel des compilateurs et interpréteurs» à la page 48.

Syntaxe de la commande LOAD

LOAD fn ... [(options)]

Options propres au VM/XA :

AMODE  $\left\{ \begin{array}{c} 24 \\ 31 \\ \text{ANY} \end{array} \right\}$  RMODE  $\left\{ \begin{array}{c} 24 \\ \text{ANY} \end{array} \right\}$

Autres options utiles :

START  $\left\{ \begin{array}{c} \text{RLDsave} \\ \text{NORLDSav} \end{array} \right\}$

Figure 90. Commande CMS LOAD

- AMODE définit le mode d'adressage lors du chargement. Il peut être différent de celui attribué par le compilateur.
- RMODE définit le mode de résidence lors du chargement. Il peut aussi être différent de celui attribué par le compilateur.

En VM/XA, la mémoire d'une machine virtuelle est décomposée en deux zones : celle qui s'étend de 0 à 16Mbytes et qui est adressable en mode 24 bits; celle qui se situe au dessus de la ligne des 16Mbytes et qui est adressable en mode 31 bits.

Un programme AMODE 24 ne peut être chargé qu'en dessous de la ligne des 16Mbytes; un programme AMODE 31 ou ANY peut être chargé en dessous ou au dessus.

Le **mode de résidence** d'un programme désigne l'endroit, par rapport à la ligne des 16Mbytes, où le programme est chargé pour l'exécution.

Un programme a un **attribut RMODE** qui peut être 24 ou ANY :

- RMODE 24 signifie que le programme ne peut être exécuté qu'en dessous de la ligne des 16Mbytes, quel que soit son AMODE.
- RMODE ANY signifie que, dans une machine XA de plus de 16M de mémoire, le programme peut être exécuté en dessous ou au dessus de la ligne des 16 Mbytes (dans ce dernier cas, AMODE ne peut être 24). S'il y a suffisamment de mémoire au dessus de la ligne il sera chargé au dessus, sinon il sera chargé en dessous.

Remarques :

1. Par défaut, le mode d'adressage est 31 si RMODE ANY est spécifié; sinon il est défini par l'attribut AMODE des programmes objets à charger.
2. Par défaut, le mode de résidence est défini par le RMODE le plus restrictif des programmes objets à charger.
3. AMODE 24 et RMODE ANY sont incompatibles.

- START déclenche l'exécution du programme directement après son chargement.

Dans une machine 370, le programme est toujours chargé en dessous de la ligne des 16 Mbytes et si START est spécifié, il sera bien entendu aussi exécuté en dessous de cette ligne. On peut toutefois charger un programme en vue de créer un module (voir «Génération d'un module»); dans ce cas le module aura les attributs AMODE et RMODE définis au niveau du LOAD et, éventuellement, ce module pourra être exploité plus tard dans une machine XA.

- RLDsave est utile si l'on a l'intention de générer un module : RLDsave permettra de générer un module **relocatable**. Lors de l'exécution, un module relocatable est chargé à une adresse appropriée à son RMODE. Un module non relocatable (option NORLDsav) est toujours chargé à une adresse fixe qui est celle à laquelle il a été généré. Cela signifie, par exemple, qu'un module non relocatable qui a été constitué dans une machine 370 (ou dans une machine XA ayant moins de 16 Mbytes de mémoire virtuelle) ne pourra jamais être exécuté au dessus de la ligne des 16Mb; un module relocatable bien, cela dépendra de son RMODE. Il y a donc toujours intérêt à ce que les modules générés à partir de programmes dont le RMODE est ANY soient relocatables <sup>12</sup>.

La "map" produite par le "loader" indique quels sont les attributs AMODE et RMODE de chaque routine.

Lorsqu'un programme a été chargé la commande PROGMAP permet de voir les adresses de son point d'entrée et de son origine, sa longueur, son attribut AMODE et s'il est relocatable ou non :

```
PROGMAP
```

Figure 91. Commande CMS PROGMAP

### 5.7.3 Génération d'un module

Lorsqu'un programme a été chargé au moyen de la commande LOAD, on peut créer un module par la commande suivante :

```
Genmod fn [MODULE [fm [(options)]]]
```

Options propres au VM/XA :

```
AMODE { 24 } RMODE { 24 }
      { 31 }          { ANY }
```

Autres options utiles en VM/XA :

```
{ STR }
{ NOSTR }
```

Figure 92. Commande CMS GENMOD

- AMODE définit le mode d'adressage. L'AMODE du GENMOD l'emporte sur l'AMODE du LOAD.
- RMODE définit le mode de résidence. Le RMODE du GENMOD l'emporte sur le RMODE du LOAD.
- Par défaut, les attributs AMODE et RMODE d'un module sont ceux définis au LOAD sauf si RMODE ANY est spécifié; dans ce cas, l'AMODE sera 31.
- STR indique que la mémoire occupée par un programme précédent doit être libérée avant le chargement du module. L'option par défaut est NOSTR (alors qu'elle était STR dans les versions précédentes du CMS). Pour créer un module compatible avec les précédentes versions de CMS, il faut spécifier STR <sup>13</sup>.

Pour connaître les caractéristiques d'un module (point d'entrée, origine, longueur, mode d'adressage, relocation), la commande PROGMAP peut être utilisée (voir Figure 91); il faut toutefois que le module soit préalablement chargé par la commande suivante :

```
LOADMOD fn [ MODULE [ fm]]
```

Figure 93. Commande CMS LOADMOD

<sup>12</sup> Il y a une autre raison pour laquelle il est conseillé de travailler avec des modules relocatables en XA : dans une machine XA de plus de 16M de mémoire virtuelle, la "page allocation table" (PAT) est située juste au dessus de la ligne des 16M. Un module non relocatable est conçu de manière à être chargé juste au dessus de la PAT. Or la dimension de cette PAT est fonction de la taille de la machine virtuelle. Dès lors, si l'on exploite un module non relocatable dans une machine XA dont la taille est supérieure à celle qui l'a généré, le module va chevaucher la PAT.

<sup>13</sup> Il faudra également exécuter la commande SET LOADAREA 20000 avant de faire le LOAD. Pour la description de cette commande, voir *CMS Command Reference, VM/XA SP*.

## 5.7.4 Exécution d'un programme

Avant d'exécuter un programme, il faut identifier les fichiers qu'il utilise par la commande FILEDEF en lui associant les 3 options suivantes (si elles ne sont pas spécifiées dans le programme):

**RECFM** définit le format des enregistrements (F, FB, FBA, FA, FS, FBS, V, VB, VBA, VA, VS, VBS). Avec VS et VBS, il faut mettre le fichier en 'OS simulated data set': fm-number 4

**LRECL** définit la longueur des enregistrements

**BLOCK (ou BLKSIZE)** définit la longueur des blocs d'enregistrements.

Exemple:

```
FI FT01F001 DISK TEST DATA (RECFM F LRECL 80 BLOCK 800)
```

Après avoir exécuté un programme, les définitions introduites par la commande FILEDEF sont abandonnées, excepté pour les programmes assembleurs. Pour les rendre permanentes, il faut ajouter l'option PERM.

Exemple:  
FI 1 DISK TEST DATA (RECFM F LRECL 50 BLOCK 500 PERM)

L'exécution d'un programme chargé par l'une des commandes LOAD ou LOADMOD est déclenchée par la commande START :

```
START [ [ { entry } [ arguments ] ] ] [(AMODE { 24  
* 31  
ANY } ] ]
```

Figure 94. Commande CMS START

- *entry* définit le point d'entrée du programme.
- \* désigne le point d'entrée par défaut;
- *arguments* désigne une liste d'arguments qui seront transmis au programme. La nature de ces arguments et la façon de les spécifier peut dépendre du langage de programmation utilisé.
- AMODE redéfinit le mode d'adressage.
- Le mode de résidence est évidemment défini au moment du chargement.

**Remarque :** Un module qui a été créé avec l'option NOMAP de la commande GENMOD ne peut être exécuté avec la commande START.

Pour exécuter directement un module, il suffit de taper le nom de ce module :

```
fn [ arguments ]
```

Figure 95. Exécution d'un module

Un module relocatable est chargé à une adresse qui dépend de son RMODE. Un module non relocatable est chargé à l'adresse définie lors du LOAD, quel que soit son RMODE.

Dans une machine 370, le programme est toujours chargé et exécuté en dessous de la ligne des 16 Mbytes et le mode d'adressage est forcé à 24.

## 5.7.5 Appel des compilateurs et interpréteurs

Avant de passer en revue les particularités de chaque compilateur disponible en CMS, il faut faire les remarques suivantes:

- Au SEGI, les compilateurs ne sont accessibles qu'en CMS batch (cf. «CMS BATCH» à la page 53).
- Tous les interpréteurs sont disponibles en CMS interactif.
- Le chargement dynamique de modules dans un programme d'applications exige que ces modules soient mis dans une TXTLIB (donc sous la forme objet), rendue accessible par un GLOBAL, ou dans un fichier TEXT dont le fn est identique au nom spécifié dans la macro LOAD ou LINK.
- Le multitasking et l'indexé séquentiel (ISAM) n'existent pas en CMS.

Voici la liste des compilateurs disponibles sous CMS:

1. VS FORTRAN
2. VS PASCAL
3. VS COBOL
4. PL/1
5. C
6. ASSEMBLER H <sup>14</sup>

Le seul interpréteur disponible est APL.

Le SEGI tient à la disposition des utilisateurs toutes les brochures décrivant les compilateurs et interpréteurs disponibles (cf. «Bibliographie» à la page 113).

<sup>14</sup> Deux ASSEMBLER sont disponibles au SEGI, l'ASSEMBLER F (accessible via la commande Assemble) et l'ASSEMBLER H (accessible via la commande HASM). Ce dernier doit être préféré car il est le seul à pouvoir générer des programmes objets compatibles XA.

### 5.7.5.1 VS FORTRAN

Le compilateur VS FORTRAN est appelé avec la commande

```
FORTVS2 fn [ (options) ]
```

Figure 96. Commande EXEC FORTVS2

Remarque: fn est le filename du fichier source dont le ft doit être FORTRAN, et 'options' représente les options du compilateur séparées par un blanc.

#### CHARGEMENT :

Le VS FORTRAN permet deux modes de chargement différents :

- en **link mode**, les routines de bibliothèques sont chargées au moment du LOAD; elles feront donc partie intégrante du load module; les commandes pour charger en link mode sont les suivantes :

```
GLOBAL TXTLIB [...] VSF2LINK VSF2FORT [...]
LOAD ...
```

- en **load mode**, les routines de bibliothèques ne sont pas chargées au moment du LOAD; elles le seront au moment de l'exécution à partir d'une bibliothèque de type LOADLIB; les commandes pour travailler en load mode sont les suivantes :

```
GLOBAL TXTLIB [...] VSF2FORT [...]
LOAD fn [...]
```

(il faut donc éviter de spécifier la bibliothèque VSF2LINK dans le GLOBAL TXTLIB) et à l'exécution :

```
GLOBAL LOADLIB VSF2LOAD
START (ou appel d'un module)
```

Dans une machine XA, pour qu'un programme FORTRAN puisse résider au dessus de la ligne des 16M, il doit avoir les attributs AMODE 31 ou ANY et RMODE ANY et il **doit être chargé en load mode** <sup>15</sup>.

#### COMMON DYNAMIQUES :

Un COMMON dynamique est un COMMON nommé dont la mémoire est réservée en dehors du load module. Son intérêt est donc de réduire la taille du module.

Exemple :

```
COMMON /COM1/ X(1000,1000,1000)
COMMON /COM2/ Y(500000)
COMMON /COM3/ Z(100,100,100)
```

Pour que COM1 et COM2 soient dynamiques, spécifier l'option DC(COM1,COM2)

- lors de l'appel du compilateur dans la commande FORTVS2; exemple :

```
FORVS2 fn (DC(COM1,COM2) [...])
```

- ou dans l'instruction @PROCESS qui précède chaque unité de programme où ces COMMON sont définis; exemple :

```
@PROCESS DC(COM1,COM2)
```

Dans une machine XA de plus de 16M, obtenir de la mémoire pour un COMMON dynamique au dessus ou en dessous de la ligne des 16M dépend du mode d'adressage du programme (son mode de résidence n'importe pas) : s'il est de 24 bits, le COMMON sera implanté en dessous; s'il est de 31 bits, le COMMON sera implanté au dessus.

#### VECTORISATION :

Sur le superordinateur, qui dispose de processeurs vectoriels, les programmes FORTRAN peuvent être vectorisés. Pour cela, il faut spécifier l'option VECTOR lors de l'appel du compilateur. Exemple :

```
FORTVS2 fn (VECTOR)
```

L'exécution d'un programme vectorisé n'est autorisée qu'en CMS batch.

Le SEGI dispose de deux bibliothèques de sous-programmes vectorisés qui peuvent être appelés par un programme FORTRAN. Il s'agit de la bibliothèque ESSL d'IBM ("Engineering and Scientific Subroutine Library") et de la "NAG FORTRAN Library" du "Numerical Algorithm Group". Pour qu'un programme FORTRAN puisse accéder à ces bibliothèques, leur nom doit être spécifié dans la commande GLOBAL. Exemple :

```
GLOBAL TXTLIB [VSF2LINK] VSF2FORT ESVVLIBA ESVVLIBB
NAGVLIB1 NAGVLIB2 CMSLIB
```

Il existe aussi une version scalaire de chacune de ces deux bibliothèques dont les noms sont

- ESVSLIBA, ESVSLIBB au lieu de ESVVLIBA, ESVVLIBB pour ESSL,
- NAGSLIB1, NAGSLIB2 au lieu de NAGVLIB1, NAGVLIB2 pour NAG.

<sup>15</sup> Cela est dû au fait que certaines routines de bibliothèques (routines d'I/O) ont un mode d'adressage de 24 bits. En load mode, ces routines, n'étant pas dans le load module, peuvent être chargées en dessous de la ligne des 16M même si le programme est chargé au dessus.

La librairie ESSL d'IBM est décrite dans la brochure suivante :

"Engineering and Scientific Subroutine Library, Guide and Reference", IBM, sc23-0184.

La librairie NAG est décrite sommairement dans le présent guide; voir «La NAG FORTRAN Library.» à la page 92.

### PARALLELISATION :

Depuis sa version 2.5.0, le compilateur VS FORTRAN permet non seulement la vectorisation, mais aussi la parallélisation des programmes. Un programme parallélisé peut exploiter les deux processeurs scalaires et les deux processeurs vectoriels du superordinateur.

Pour qu'un programme soit parallélisé, il faut spécifier l'option PARALLEL (avec ses sous-options éventuellement) lors de l'appel du compilateur:

```
FORTVS2 fn (PARALLEL
```

Les modules relatifs au parallélisme se trouvent dans la librairie VSF2PARA TXTLIB qui doit être spécifiée, en plus des librairies habituelles, dans la commande GLOBAL TXTLIB précédant le LOAD. Il ne faut pas spécifier la librairie VSF2LINK; le chargement doit obligatoirement se faire en *load mode*.

L'exécution en parallèle n'est autorisée qu'en CMS batch. De plus, elle n'est possible qu'à partir d'un module relocatable. Il faut donc spécifier l'option RLD dans la commande LOAD et générer ensuite un module au moyen de la commande GENMOD.

Pour lancer l'exécution du module, il faut spécifier la *run-time option* PARALLEL(n) où n est le nombre de processeurs virtuels. Avec 2 processeurs réels, n doit être au moins égal à 3 car un des processeurs virtuels est réservé pour le CMS. D'autre part, les règles d'exploitation en vigueur au SEGI imposent une limite supérieure au nombre de processeurs virtuels en fonction des classes (voir «Tableau des limitations par classe (travaux batch)» à la page 85).

### RESUME :

Pratiquement, on procédera comme suit :

1. Compilation - elle peut se faire dans une machine 370 ou XA; le compilateur FORTRAN produit un module objet AMODE ANY et RMODE ANY :

```
FORTVS2 fn [(options]
```

Les options VECTOR et/ou PARALLEL, et leurs sous-options éventuelles, doivent notamment être spécifiées si le programme doit être vectorisé et/ou parallélisé.

2. Exécution sans passer par un module (si le programme est parallélisé, ce mode d'exécution ne convient pas; passer au point suivant) - elle doit se faire dans une machine XA de plus de 16M de mémoire virtuelle si l'on veut profiter pleinement de l'architecture XA :

```
GLOBAL TXTLIB [...] VSF2FORT [...] CMSLIB 16  
LOAD fn  
GLOBAL LOADLIB VSF2LOAD  
FILEDEF ...  
START
```

3. Génération d'un module - elle peut se faire dans une machine 370 ou XA; l'option RLD de la commande LOAD permet de créer un module relocatable (les modules qui doivent être chargés au dessus de la ligne des 16M et les modules parallélisés doivent être relocatables) :

```
GLOBAL TXTLIB [...] [VSF2PARA] VSF2FORT [...] CMSLIB 16  
GLOBAL LOADLIB VSF2LOAD  
LOAD fn [(RLD)  
GENMOD fn
```

La librairie VSF2PARA doit être spécifiée si le programme est parallélisé.

4. Exécution du module - elle doit se faire dans une machine XA de plus de 16M de mémoire virtuelle si l'on veut profiter pleinement de l'architecture XA; l'option PARALLEL(n) doit être spécifiée si le programme est parallélisé :

```
GLOBAL LOADLIB VSF2LOAD  
FILEDEF ...  
fn [PARALLEL(n)]
```

n désigne le nombre de processeurs virtuels.

**Remarque :** Le SEGI dispose également du VAST-2, préprocesseur pour l'optimisation, la vectorisation et la parallélisation des programmes FORTRAN (voir «Bibliographie» à la page 113).

### 5.7.5.2 VS PASCAL

Le compilateur VS PASCAL est appelé par la commande

---

<sup>16</sup> Dans tous les cas, il est conseillé de faire le chargement en *load mode*; pour cela, il ne faut pas spécifier VSF2LINK. C'est obligatoire pour les programmes de plus de 16M et pour les programmes parallélisés.

```
VSPASCAL fn [ (options)]
```

**Figure 97. Commande EXEC VSPASCAL**

fn est le filename d'un fichier dont le filetype est PASCAL et qui contient le programme source. Cette commande crée un module objet dans un fichier fn TEXT.

La commande suivante permet de créer un module exécutable à partir de modules objets fournis par le compilateur :

```
PASCMOD fn1 [fn2 ...] [(options)]
```

**Figure 98. Commande EXEC PASCMOD**

Pour une description complète de l'utilisation du compilateur VS PASCAL, voir la documentation IBM («Bibliographie» à la page 113).

### 5.7.5.3 VS COBOL

Le compilateur VS COBOL peut être appelé par la commande

```
COBOL2 fn [( options )]
```

**Figure 99. Commande EXEC COBOL2**

fn est le filename d'un fichier dont le filetype est COBOL et qui contient le programme source. Cette commande crée un module objet dans un fichier fn TEXT.

Chargement :

```
GLOBAL TXTLIB [...] VSC2LTX [ ... ]  
LOAD fn [ ... ]
```

Exécution :

```
GLOBAL LOADLIB VSC2LOAD  
START (ou appel d'un module)
```

Pour des détails, voir la documentation IBM («Bibliographie» à la page 113).

### 5.7.5.4 PL/1

Le compilateur PL/1 peut être appelé avec la commande

```
PLIOPT fn [(options)]
```

**Figure 100. Commande EXEC PLIOPT**

fn est le filename du programme source dont le ft doit être PLIOPT ou PLI.

Pour le chargement, les librairies PLILIB et IBMLIB doivent être accessibles. En général, on utilisera les commandes suivantes :

```
GLOBAL TXTLIB PLILIB CMSLIB IBMLIB  
LOAD fn (options)
```

Pour créer un module, il est nécessaire de spécifier l'option FROM PLISTART :

```
GENMOD fn (FROM PLISTART)
```

A l'exécution, les librairies PLILIB et IBMLIB sont aussi nécessaires. Donc si l'exécution (d'un module par exemple) est séparée du chargement, il faut de nouveau rendre les librairies accessibles par

```
GLOBAL TXTLIB PLILIB CMSLIB IBMLIB
```

Exemple : pour compiler et exécuter un programme PL/1 qui utilise le fichier introduit par le ddname FILE1, qui lit des données au terminal (fichier SYSIN) et écrit sur l'écran (fichier SYSPRINT), on fera:

```
PLIOPT TEST (= compiler programme PLI situé dans  
le fichier TEST PLIOPT)  
GLOBAL TXTLIB PLILIB CMSLIB IBMLIB  
LOAD TEST  
FI FILE1 DISK TEST DATA (PERM RECFM FB LRECL 80 BLOCK 800  
FI SYSIN TERM  
FI SYSPRINT TERM  
START
```

**Remarque :** En mode XA, les variables automatiques de PL/1 sont toujours allouées en dessous de la ligne des 16M. Les variables CONTROLLED et BASED, par contre, peuvent être allouées en dessous au au dessus de la ligne des 16M à condition que la sous-option ANYWHERE de la "run-time option" HEAP soit active (c'est l'option par défaut).

### 5.7.5.5 C

Le compilateur C peut être appelé avec la commande

```
CC fn [ ft [ fm ]] [ ( options )
```

**Figure 101. Commande EXEC CC**

Remarques :

1. fn, ft, fm désignent respectivement le filename, le filetype et le filemode du programme source. Le filetype par défaut est C. le filemode par défaut est A.
2. Le compilateur C distingue les majuscules et les minuscules.

La commande suivante permet de créer un module exécutable à partir du fichier TEXT fourni par le compilateur :

```
CMOD fn
```

**Figure 102. Commande EXEC CMOD**

Les bibliothèques nécessaires à l'exécution d'un module doivent être définies comme suit :

```
GLOBAL TXTLIB EDCBASE IBMLIB CMSLIB  
GLOBAL LOADLIB EDCLINK
```

**Remarque :** Le langage C utilise des caractères spéciaux qui, lorsqu'ils sont entrés sur un terminal "table 500"<sup>17</sup> ne génèrent pas le code hexadécimal attendu par le compilateur (correspondant à l'EBCDIC étendu), ce qui provoque des erreurs de syntaxe à la compilation. Les caractères qui posent problème en C sont le point d'exclamation, les crochets ouvert et fermé et la barre verticale.

Pour créer ou modifier un programme C sur un terminal "table 500", il faut changer les tables de traduction utilisées par CMS, de façon à assigner à tous les caractères utilisés le code hexadécimal requis. Ceci assure également la portabilité du programme. C'est l'objectif de la commande suivante :

```
TRAN5TOE
```

**Figure 103. Commande EXEC TRAN5TOE**

Cette commande modifie les tables de traduction du CMS pour les 8 caractères qui génèrent un code hexadécimal différent de l'EBCDIC étendu, lorsqu'on les entre sur un

terminal "table 500". Elle permet l'entrée des caractères spéciaux ainsi que leur affichage correct au terminal.

La commande suivante annule l'effet de TRAN5TOE :

```
TRAN5OFF
```

**Figure 104. Commande EXEC TRAN5OFF**

Pour une description complète de l'utilisation du compilateur C, voir la documentation IBM («Bibliographie» à la page 113).

### 5.7.5.6 ASSEMBLER H

L'assembleur H est appelé avec la commande

```
HASM fn [(options)]
```

**Figure 105. Commande CMS HASM**

Remarque: fn est le filename du programme dont le ft doit être ASSEMBLE. 'options' représente les options de l'assembleur qui peuvent être obtenues en faisant HELP ASSEMBLE.

Si le programme ASSEMBLER fait appel à des macro-instructions, il faut définir les bibliothèques qui contiennent ces macro-instructions au moyen d'une commande GLOBAL MACLIB avant d'appeler l'ASSEMBLER. Le système CMS possède les bibliothèques publiques de macro-instructions suivantes :

DMSOM, DMSGPI qui contiennent les macros CMS;  
OSMACRO, OSPSI, OSVSAM qui contiennent les macros OS simulées sous CMS.

### 5.7.5.7 APL

La commande pour entrer dans l'environnement APL est

```
APL2
```

**Figure 106. Commande EXEC APL2**

<sup>17</sup> Appartiennent à la catégorie "table 500", tous les terminaux de marque IBM installés à l'Université. Les micro-ordinateurs raccordés en mode 7171 (éventuellement au travers des PBX) appartiennent à la catégorie "table 500" si le type de terminal sélectionné lors de la connexion contient les caractères "500" (ex.: IBM500 ou IBM500R). Pour eux, il sera donc plus judicieux de sélectionner le fonctionnement en EBCDIC étendu (ex.: types IBM ou IBMR), ne demandant aucune conversion supplémentaire. Pour plus de détails, voir «Annexe B. Les jeux de caractères employés au SEGI» à la page 121.

Celles qui permettent de quitter l'environnement APL sont:

```
)OFF [HOLD]
)CONTINUE [HOLD]
```

**Remarque :** l'APL utilise des caractères spéciaux. Il n'est donc utilisable que sur les terminaux possédant les caractères APL.

## 5.8 Sauvegarde de l'environnement VM.

Lors de l'exécution de programmes ou de procédures de commandes EXEC, on est souvent amené à modifier certaines caractéristiques de l'environnement VM, par exemple l'accessibilité de certaines TXTLIB, "HOLD" du printer, etc.

Pour permettre de rétablir aisément ces dispositifs à l'état initial, le SEGI a mis au point quatre procédures de commandes (EXEC):

**SPPUSH et SPPOP** permettent de mémoriser et restaurer l'état des SPOOL.

**GLPUSH et GLPOP** permettent de conserver les GLOBAL effectués pour les MACLIB, TXTLIB et LOADLIB.

### 5.8.1.1 Utilisation de SPPUSH et SPPOP.

Syntaxe:

```
SPPUSH dev
SPPOP dev
```

**Figure 107. Commande EXEC SPPUSH ET SPPOP**

Remarque: dev peut prendre la valeur READER, PRINTER ou PUNCH.

Exemple:

On désire imprimer plusieurs petits fichiers en "continu" (sans séparation entre eux); la séquence des commandes sera:

```
SPPUSH PRINTER
CP SPOOL PRINTER CONT
PRINT UN DATA
PRINT DEUX DATA
... ..
```

rétablissement du printer en l'état antérieur par:

```
SPPOP PRINTER
```

### 5.8.1.2 Utilisation de GLPUSH et GLPOP.

Syntaxe:

```
GLPUSH lib
GLPOP lib
```

**Figure 108. Commande EXEC GLPUSH AND GLPOP**

Remarque: lib peut être MACLIB ou TXTLIB ou LOADLIB. Exemple: Un utilisateur travaillant normalement en FORTRAN veut utiliser momentanément un programme PL/I; la séquence sera:

```
GLPUSH TXTLIB
GLOBAL TXTLIB PLILIB CMSLIB IBMLIB
LOAD du programme PLI
GLPOP TXTLIB
```

GLPOP TXTLIB rétablit le GLOBAL préexistant.

## 5.9 CMS BATCH

### Notions générales

En CMS, les travaux batch sont exécutés dans des machines déconnectées réservées à ce seul usage. Ces machines exécutent tous les travaux qu'elles reçoivent dans leur reader. Pour soumettre un travail batch, il suffit donc de l'envoyer dans le reader d'une machine batch ce qui peut être fait comme suit:

Préparer un fichier ayant la structure indiquée ci-dessous puis l'envoyer à la machine batch via la commande PUNCH (cf infra)

```
carte/JOB
      .
      .   travail à exécuter
      .
/*
```

La carte /JOB a le format suivant:

```

/JOB userid acctnumber jobname [CLASS { A
class } ] [TAPES { 0
n } ] [CAS { 0
n } ]
[CAS3490 { 0
n } ] [ { HOLD
NOHOLD } ] [TDISK { nnnnnn
MAX } ] [HOUR hh]
[SUCCES jobname][ PW logpassword]

```

**Figure 109. LA CARTE /JOB**

Remarque: 'userid' est le nom de la machine virtuelle à qui la machine CMS BATCH enverra les messages permettant de savoir quand le job démarre et quand il se termine, c'est-à-dire:

```

JOB 'jobname' STARTED
JOB 'jobname' ENDED ou JOB 'jobname' ABEND

```

En plus ce 'userid' et le code destination qui lui est associé apparaîtront sur les listings produits par le travail batch (en vue de la restitution). Enfin, tous les fichiers SPOOL créés par le job batch auront ce "userid" pour caractéristique ORIGINE. La faculté de pouvoir soumettre un job avec un "userid" différent de celui de la machine soumettant le job est soumise à une autorisation par ACF2. Idem pour pouvoir soumettre à partir d'un noeud extérieur.

'acctnumber' est le numéro d'accounting sur lequel sera facturé le travail.

'jobname' est le nom du travail (maximum 8 caractères).

**CLASS** indique la classe de passage du travail. Si le paramètre est codé, il aura priorité sur la classe dans laquelle le travail aura été envoyé (via le punch en particulier).

**TAPES** indique le nombre de dérouleurs de bandes magnétiques nécessaires pour exécuter le travail.

**CAS** indique le nombre de dérouleurs de cassettes nécessaires pour exécuter le travail.

**CAS3490** indique le nombre de dérouleurs de cassettes de type 3490 nécessaires pour exécuter le travail.

**HOLD** indique que le job ne sera exécuté qu'après sa libération via la commande BATCH RELEASE (cfr. plus loin) ou via la fin d'un job prédécesseur dans une chaîne de jobs.

**TDISK nnnn** indique l'espace DISK **temporaire** exigé pour une exécution normale du job. Sur le superordinateur, le maximum est de 350 cylindres de disque 3390. Le job ne démarre que si l'espace TDISK demandé est disponible. Cependant, il est conseillé de le réserver et de le formater dès le début du job (Via EXEC TDSKCMS OU EXEC TDSKOS) sinon il y a risque que cet espace TDISK soit entretemps acquis par une autre machine virtuelle.

Les jobs de classe G ou H peuvent aussi spécifier TDISK MAX, auquel cas le job disposera d'un minidisque supplémentaire préformaté à l'adresse 200 d'une capacité de 2654 cyl. 3390 (soit 1866 Mb au total)

**HOUR hh** indique l'heure après laquelle le job pourra s'exécuter. Cette heure est celle se présentant APRES la soumission du job. Exemple : un job soumis à 12h15 avec le paramètre HOUR 12 ne pourra pas s'exécuter avant le lendemain midi.

**SUCCES jobname** indique le nom du job successeur qui est à libérer après l'exécution de celui-ci, à condition que cette exécution ne se termine pas par un double BATCH CANCEL. Tous les jobs dont le nom est cité sont libérés à condition qu'ils appartiennent au même propriétaire que le job qui se termine. Un job successeur ne peut s'appeler "NO" car il ne sera jamais libéré.

**PW logpassword** indique le mot de passe associé au userid spécifié dans la carte /JOB. Ce mot de passe est obligatoire si le job est d'origine extérieure à BLIULG11.

Remarques :

1. Les paramètres de la carte/JOB doivent être contenus dans les 71 premiers bytes. Si cette zone est insuffisante, on peut indiquer en col. 72, par un caractère non blanc, la présence d'une suite de carte/JOB. Celle-ci sera de la forme :

```

/ en col. 1
blanc en col. 2
paramètres en col. 3 à 71

```

2. On peut faire suivre la carte /JOB par une carte /SET pour limiter les ressources en CPU et en lignes imprimées que peut s'octroyer le travail batch. Le format de la carte /SET est:

```

/SET TIME seconds PRINT lines

```

**Figure 110. CARTE /SET TIME**

Exemple:

```

/SET TIME 5 PRINT 5000

```

pour limiter le CPU à 5 secondes et le nombre de lignes à 5000. A défaut de carte /SET, les limites en CPU et en lignes sont les limites maximales de la classe du job (cfr tableau des classes). Toute carte /SET ne suivant pas immédiatement la carte /JOB est ignorée.

Sous CMS, on travaille en batch de la même manière qu'en interactif (pas de JCL comme en OS/VS2). Les trois seules différences sont:

1. En batch, CMS lit les commandes à exécuter dans le reader au lieu de les lire au terminal comme cela se fait en interactif: on doit donc mettre dans le reader de

la machine batch toutes les commandes que l'on désire exécuter comme on le ferait au terminal excepté que les commandes CP doivent être précédées du libellé 'CP'.

2. En standard la machine batch a accès aux disques suivants:

- 195 A disque A de la machine batch à ne pas confondre avec le disque A de la machine qui a envoyé le job.
- 192 D (logiciels publics)
- 190 S (logiciels publics)
- 19E Y (logiciels publics)

Donc si l'on désire que la machine batch ait accès à d'autres disques, et notamment aux disques de la machine qui a envoyé le travail, il faudra établir les connexions nécessaires.

3. Les informations qui, en interactif, apparaissent au terminal, sont, en batch, envoyées dans le 'listing console'. Ce listing console est envoyé dans le reader de 'userid' dès la fin du job batch. Il est possible de donner une autre destination au listing console en faisant:

```
CP TAG DEV CONS PRLx
CP SPOOL CONS RSCS
```

pour l'imprimer sur une imprimante remote PRLx ou

```
CP SPOOL CONS FOR *
```

pour l'imprimer sur l'imprimante locale du SEGI.

Les autres paramètres de la commande CP SPOOL peuvent évidemment s'appliquer également au listing console (DIST,HOLD,etc...). Ainsi, si l'utilisateur de la machine USER1 désire compiler et exécuter le programme PL/I TEST qui se trouve sur son disque 191, il devra effectuer les opérations suivantes:

1. Créer un fichier, par exemple TEST JOB, qui contient:

```
/JOB USER1 acctnumber JOB1
CP LINK * 191 191 RR
AC 191 B
PLIOPT TEST
GLOBAL TXTLIB PLILIB CMSLIB IBMLIB
LOAD TEST
START
/*
```

2. Envoyer le fichier TEST JOB sur le reader de la machine BATCH en faisant:

```
SPOOL PU BATCH CL E FORM OFF CL E
PUNCH TEST JOB (NOH
```

afin que la classe de passage soit E.

Remarque:

Dans ce cas, les fichiers créés par PLIOPT sur le disque A de la machine batch sont perdus. Pour les récupérer, il faut les renvoyer dans le reader de votre machine par exemple en utilisant la commande SENDFILE fn ft fm. Ils seront de plus taggés avec des informations telles que : nom de la machine batch qui a exécuté le job, nom du job, date et heure du renvoi. L'exemple deviendra donc:

```
/JOB USER1 acctnumber JOB1
CP LINK * 191 191 RR
AC 191 B
PLIOPT TEST
SF TEST TEXT TO USER1
SF TEST LISTING TO USER1
GLOBAL TXTLIB PLILIB CMSLIB IBMLIB
LOAD TEST
START
/*
```

### 5.9.1.1 Exec commandes SENBATCH et RECBATCH

<sup>18</sup> Pour simplifier l'utilisation du batch, le SEGI a développé une procédure EXEC appelée SENBATCH qui, automatiquement:

1. Génère la carte /JOB
2. Génère l'appel de l'EXEC JOBHEADE qui se chargera de
  - faire un GLOBAL MACLIB CMSNEW DMSGPI OSMACRO OSMACRO1
  - faire un GLOBAL TXTLIB PLILIB IBMLIB VSF2LINK VSF2FORT VSF2MATH VSC2LTXC CMSLIB TSOLIB
  - faire un GLOBAL LOADLIB VSC2LOAD OSVSLOAD
  - établir la connexion en lecture vers le disque 191 de la machine propriétaire avec l'adresse virtuelle 191 et le filemode B/A.
3. Permet à l'utilisateur d'entrer la liste des commandes à exécuter dans le job (on termine la liste en entrant une ligne nulle).
4. Génère l'appel de l'EXEC JOBTRAIL qui se chargera de renvoyer dans le reader de la machine propriétaire du job **tous** les fichiers que celui-ci a créés sur le disque A de la machine batch ainsi qu'un listing console qui est une image de ce que l'on aurait eu au terminal si l'on avait exécuté ce travail en interactif.
5. Envoie le job ainsi constitué dans le reader de la machine BATCH.

<sup>18</sup> Dans la description du système BATCH nous conviendrons d'appeler "machine émettrice" celle qui a soumis le JOB, et "machine propriétaire" celle dont le USERID est codé dans la carte JOB.

L'exec commande SENBATCH a la syntaxe suivante:

```
{ SENBATCH } [Jobname jobname] [Userid userid] [Number accountnumber]
  SB
  [Input filename] [Class class] [Tapes n] [CAS n] [CAS3490 n]
  [Password password] [Seconds n] [Lines n]
  [AT noeud] [Hold] [TDISK n|MAX] [HOUR hh]
  [SUCCES jobname]
  [PW logpassword]
```

Figure 111. COrmande EXEC SENBATCH

Remarque:

<b>jobname</b>	nom du job
<b>userid</b>	nom de la machine propriétaire (ce USERID doit être défini sur l'ordinateur sur lequel est envoyé le JOB)
<b>accountnumber</b>	numéro de facturation (ce numéro doit être défini sur l'ordinateur sur lequel est envoyé le JOB)
<b>filename</b>	nom d'un fichier de type JOB dont le contenu remplacera la lecture au terminal
<b>class</b>	classe de passage
<b>Tapes n</b>	nombre d'armoires à bandes
<b>CAS n</b>	nombre d'armoires à cassettes 3480
<b>CAS3490 n</b>	nombre d'armoires à cassettes 3490
<b>password</b>	est le mot de passe en lecture associé au disque 191 de userid. Si le job est exécuté sur le même ordinateur et que le soumetteur est le userid précisé dans la commande, ce paramètre peut être omis.
<b>Seconds n</b>	limite CPU exprimée en secondes
<b>Lines n</b>	limite impression exprimée en lignes
<b>Noeud</b>	cette option n'est utilisée que pour envoyer le JOB vers un autre ordinateur du réseau. "noeud" indique l'ordinateur visé. (cfr «Réseau networking» à la page 78). Sauf indication contraire les résultats reviennent dans le reader de la machine émettrice. Si l'option "AT" est utilisée les options "Userid" et "Number" sont obligatoires.
<b>Hold</b>	pour soumettre le job mais empêcher son exécution.

Les paramètres **TDISK**, **HOUR**, **PW** et **SUCCES** ont la même signification que précédemment (Voir Figure 109 à la page 54)

Par défaut, la commande aura comme paramètres:

```
SB JOBNAME JOBx USERID userid NUMBER numéro de
l'émetteur CLASS A TAPES 0 CAS 0 TDISK 0
HOUR NO SUCCES NO
```

où

x est la classe de passage du job  
userid est le userid de la machine émettrice

On notera que le paramètre CLASS spécifié dans la commande SB supplante la classe indiquée dans une commande SPOOL préalable.

Donc, en utilisant SB, on peut réaliser l'exemple donné précédemment en faisant:

```
SB J JOB1 CL E
PLIOPT TEST
LOAD TEST
START
ligne nulle
```

On peut également réaliser ce même exemple en utilisant le paramètre INPUT comme suit:

1. Créer le fichier dont le ft sera JOB et qui contiendra les commandes à exécuter en batch:

```
PLIOPT TEST
LOAD TEST
START
```

2. Soumettre le travail batch comme suit:

```
SB J JOB1 CL E I MYJOB
```

Lorsque le job est terminé, on peut effectuer l'exec commande

```
{ RECBATCH }
  RB }
```

Figure 112. COrmande EXEC RECBATCH

RB appelle la commande RDRLIST (cfr «Lister et manipuler le contenu du reader: RDRLIST» à la page 21) pour afficher les seuls fichiers de classe B situés sur votre reader (La commande SB renvoie ses résultats en classe B) et triés dans l'ordre descendant de leur date et heure de création. On peut ainsi utiliser les fonctions standards de RDRLIST pour les afficher ou les recopier sur notre minidisque. A noter cependant que RB affecte la fonction de DISCARD (ou PURGE) à la touche fonction PF10 et la fonction RECEIVE WITH REPLACE à la touche fonction PF4.

Tous les fichiers créés par le travail batch sur le disque 195 A de la machine batch et qui n'ont pas été volontairement abandonnés par une commande ERASE peuvent ainsi être facilement récupérés.

Dans notre exemple, la commande RB va ramener dans le reader de la machine USER1 les trois fichiers:

```
TEST TEXT
TEST LISTING
JOB1 CONSOLE
```

On peut évidemment éviter de renvoyer TEST TEXT et TEST LISTING en procédant comme suit:

```
SB CL E J JOB1
PLIOPT TEST
LOAD TEST
START
ERASE TEST *
ligne nulle
```

Voici un listage des cartes générées par SENBATCH

```

                                         Colonne 72 --+
                                         |
/JOB userid acctnumber jobname TAPES n CAS n CAS3490 n* <--+
/ TDISK 0 HOUR NO SUCCESS NO
EXEC JOBHEADE password
.
.   commandes de l'utilisateur
.
EXEC JOBTRAIL
```

JOBHEADE effectue les opérations suivantes:

1. Spool de la console vers le reader de la machine propriétaire du job (userid de la carte /JOB)
2. Connexion du disque 191 de la machine propriétaire avec le fm B concaténé au disque A de la machine batch
3. GLOBAL MACLIB CMSNEW DMSGPI OSMACRO OSMACRO1
4. GLOBAL TXTLIB PLILIB VSF2LINK VSF2FORT CMSLIB IBMLIB TSOLIB

JOBTRAIL renvoie dans le reader de la machine propriétaire les fichiers créés par le job sur le disque A de la machine batch. Ces fichiers sont transmis en format NETDATA.

Remarques:

1. Les jobs n'ayant pas été envoyés au système par la commande SB (par exemple, ceux lus sur un lecteur réel ou transmis par TSO) pourront eux aussi être récupérés par la commande RB, à condition que l'utilisateur insère, en début de job, la commande

```
EXEC JOBHEADE [password]
```

**Figure 113. Commande EXEC JOBHEADE**

et, en fin de job, la commande

```
EXEC JOBTRAIL
```

**Figure 114. Commande EXEC JOBTRAIL**

Remarque: 'password' est le mot de passe associé en lecture au disque 191 de la machine référencée par le userid codé dans la carte /JOB. Il n'est requis que si l'origine du job est extérieure au noeud où s'exécute le job ou si le soumetteur est différent du userid.

2. Il est souvent très pratique de préparer des procédures EXEC pour les travaux fréquemment envoyés en batch: il suffit alors d'appeler la commande SB et de donner le nom de l'exec à exécuter suivie d'une ligne nulle. Exemple: si le fichier fn1 EXEC A1 contient:

```
PLIOPT &1
LOAD &1
START
```

on fera:

```
SB C E J JOB1
EXEC fn1 TEST
ligne nulle
```

pour compiler et exécuter le programme TEST PLI en batch.

3. Une technique utile est de pouvoir insérer dans le job proprement dit, un fichier de format quelconque destiné à être recréé et traité par le job sur disque 195-A de la machine CMSBATCH. Le format (Fixe,80) imposé aux jobs oblige à passer par le format NETDATA ce qui se fait de deux façons possibles:

- a. Lors du dialogue établi par la commande SENBATCH, il suffit d'entrer la ligne

```
GETNDATA fn ft [fm]
```

- b. Si le job existe déjà, il suffit de l'éditer, de positionner la "current line" à l'endroit qui précède le lieu d'insertion, puis d'entrer la commande (XEDIT macro)

```
GETNDATA fn ft [fm]
```

Dans les deux cas, le job aura la structure suivante :

```

/JOB USER1 acctnt jobname ...
...
CP/CMS command
...
GETNDATA
-   records de longueur 80 nécessaires à
-   la recréation du fichier qui a été
-   l'objet du GETNDATA lors de la
-   création du job
...
CP/CMS command utilisant en R/W le fichier recréé
...
/*

```

La commande GETNDATA exécutée par le job recrée le fichier grâce aux cartes données qui la suivent, avec le même "filemode number" et la même date de dernière mise à jour que le fichier original.

Si le fichier existe déjà sur 195-A, il est remplacé.

Si une erreur se produit pendant la recréation du fichier (disque 195 trop petit, par exemple), alors le job sort en CMS BATCH ABEND

Cette technique est valable pour tout fichier ne dépassant pas 55 Mb. Elle est indiquée pour toute exécution batch sur un noeud extérieur.

A la fin du job, le fichier mis à jour par le job est renvoyé au propriétaire du job grâce à la commande JOBTRAIL.

## 5.9.2 Contrôle des travaux batch en VM

<sup>19</sup> La machine de nom BATCH, créée par le SEGI, est une machine virtuelle déconnectée chargée de recevoir les travaux différés des utilisateurs et de les distribuer aux différentes machines batch (BATCHZ1, BATCHA1, BATCHD1, ...) dès que celles-ci sont disponibles. Elle facilite également le contrôle des travaux et de l'activité des machines batch par le SEGI.

Pour l'utilisateur, l'intérêt de cette machine est double; elle lui donne en effet la possibilité:

- d'obtenir des informations sur l'état des travaux qu'il a déjà transmis et, le cas échéant, de réorienter un travail;
- d'intervenir au niveau de l'exécution de ses travaux, par exemple en demandant leur suppression ("CANCEL"), leur mise en attente ("HOLD"), un transfert de classe, etc...

Ces opérations sont mises en oeuvre par des commandes que l'utilisateur adresse à la machine BATCH et dont la syntaxe générale est de la forme (il s'agit d'une procédure CMS):

Batch verbe opérandes

**Figure 115. Commande EXEC BATCH**

Les commandes disponibles, dont on obtiendra d'ailleurs le détail par la commande Help BATCH, sont les suivantes:

### 5.9.2.1 Connaître l'état d'un ou plusieurs travaux

La commande:

```

Batch Query [ Jobname jobname
             [ Number jobnumber ] ] [Class class] [For userid]
           [ NOHold ] [ Active ] [ Maximum number ]           [AT noeud]
           [ Userhold ] [ Kept ] [ Maximum * ]
           [ Syshold ] [ Waiting ] [ Maximum 20 ]

```

**Figure 116. Commande EXEC BATCH**

Remarque:

- jobname** le nom du travail; s'il est ambigu avec un des mots- clé, J est codé devant;
- jobnumber** le numéro de travail (= spoolid); ce numéro peut être employé en lieu et place du jobname;
- class** la classe des travaux recherchés;
- userid** le nom de la machine ayant soumis le travail ou pour le compte de laquelle il s'exécute; le défaut est le nom de la machine émettrice de la commande;
- NOHold** sélection des travaux sans "hold" (= maintien en attente) de l'utilisateur ni de l'opérateur
- Hold** sélection des travaux avec "hold" de l'utilisateur ou de l'opérateur
- Userhold** sélection des travaux avec "hold" de l'utilisateur
- Syshold** sélection des travaux avec "hold" de l'opérateur
- Active** sélection des travaux qui s'exécutent
- Kept** sélection des travaux à "problèmes"
- Waiting** sélection des travaux en attente d'exécution
- maximum** nombre maximum de réponses, par défaut 20 ( \* = pas de limite)
- noeud** désigne le nom du noeud de l'ordinateur qui doit exécuter la commande

<sup>19</sup> Le contrôle des travaux batch n'est possible qu'à partir de la machine propriétaire.

En réponse à cette commande, la machine BATCH fournira la liste des jobs sélectionnés et les informations suivantes:

**CLASSE/POSITION** il s'agit de la classe de dépôt et de la position "effective" dans cette classe (les travaux en "hold" n'étant pas comptés).

**DATE-ON/TIME-ON** date et heure de prise en charge (pour un travail en attente) ou de début d'exécution (pour un travail actif ou dont l'exécution a été interrompue); voir ci-après "Status".

**STATUS** un job peut être dans l'un des états suivants:

**WAITING** le job est en attente d'exécution

**NOM** d'une machine batch: le job s'exécute sur cette machine

**KEPT** cas exceptionnel probablement dû à un arrêt d'exploitation batch durant l'exécution du job; dans ce cas, l'utilisateur peut prendre contact avec le contrôle des opérations ou simplement "canceller" son job; voir infra commande batch cancel

**HOLD** l'état "hold" que l'utilisateur contrôle est indiqué par USER et est indépendant de celui dont l'opérateur a la responsabilité (indiqué par SYS).

Exemples:

```
B Q
```

liste tous les travaux soumis par la machine émettrice ou pour elle (maximum 20)

```
B Q TEST
```

donne l'état du travail 'TEST'

```
B Q H M * F U123401
```

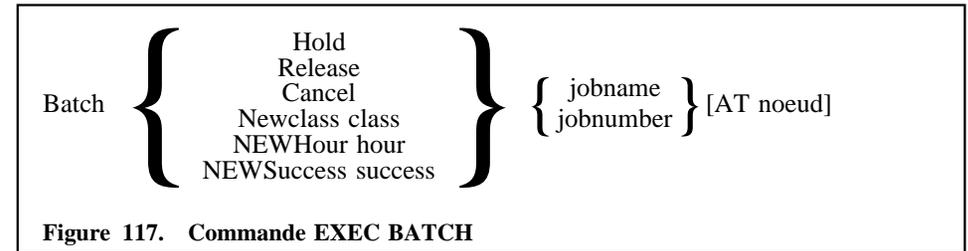
liste tous les travaux en "hold" pour la machine U123401

```
B Q C A NOH
```

donne le nombre de travaux de classe A exécutables soumis par la machine émettrice ou pour elle.

Remarque: tous les opérantes restreignent la sélection. MAXIMUM 0 indique que seul le nombre des travaux doit être renseigné.

### 5.9.2.2 *Intervenir sur l'état d'un travail donné*



**jobname/jobnumber** nom ou numéro du travail. Si plusieurs travaux ont le même nom, le numéro doit être utilisé. L'utilisateur n'a accès qu'aux travaux qu'il a soumis ou qui sont soumis pour son compte.

**Hold** met le travail en "hold"

**Release** l'inverse de Hold

**Cancel** supprime le travail du système. S'il est actif, l'exécution est arrêtée.

**Newclass class** permet de changer la classe du travail sans modifier son ordre d'arrivée.

**NEWHour hour** permet de changer l'heure à laquelle le job peut démarrer ou d'enlever cette contrainte en mettant hour = NO

**NEWSucces succes** permet de changer le nom d'un job successeur ou d'enlever ce POST-traitement en mettant succes = NO

**noeud** désigne le nom du noeud de l'ordinateur qui doit exécuter la commande.

Remarque: nous insistons auprès des utilisateurs pour qu'ils n'interviennent plus sur leurs travaux batch par la commande TRANSFER; d'une part, ce n'est pas utile, étant donné les possibilités offertes par les commandes BATCH et, d'autre part, il pourrait s'ensuivre des exécutions non désirées de travaux que l'utilisateur ne pourrait plus interrompre.

## 5.9.3 Recouvrement des résultats en cas d'abend sous CMSBATCH

Sous CMSBATCH:

en cas d'abend du job par limite de TIME, PRINT ou PUNCH,

en cas de cancel du job par la commande BATCH CANCEL,

en cas d'abend d'un programme utilisateur et même

en cas d'abend du système CMS,

il est possible de récupérer les fichiers du disque A de la machine BATCHx. Cette récupération, qui est facultative, se fait de manière similaire à l'EXEC JOBTRAIL. Le coût du traitement de ces opérations est évidemment imputé au même compte que celui du job. On dispose des deux commandes WANTDISK et WANTDUMP qui peuvent être placées n'importe où dans le job et en n'importe quelle quantité. Il faut

noter qu'un deuxième BATCH CANCEL d'un job provoque l'arrêt immédiat du job et abrège donc une éventuelle exécution de JOBTRAIL.

### 5.9.3.1 WANTDISK

Syntaxe:

```
WANTDISK [NO]
```

Figure 118. Commande CMS WANTDISK

Sans l'opérant NO, tout abend ultérieur sera suivi d'un JOBTRAIL qui renvoie les fichiers du disque A dans le "reader" de l'utilisateur. Avec l'opérant NO, on annule l'effet d'une commande WANTDISK antérieure. Au début de chaque job, l'option WANTDISK est active.

### 5.9.3.2 WANTDUMP

Syntaxe:

```
WANTDUMP [NO]
```

Figure 119. Commande CMS WANTDUMP

Sans l'opérant NO, tout abend ultérieur ne résultant pas d'un dépassement TIME, PRINT ou PUNCH, sera suivi de l'exécution de la commande "CP DUMP 0-END" qui renvoie le dump de la mémoire de la machine dans le "reader" de l'utilisateur. Avec l'opérant NO, on annule l'effet d'une commande WANTDUMP antérieure. Au début de chaque job, l'option WANTDUMP est NO.

## 5.10 Impression en format A4: imprimante 3812

### 5.10.1 Transmission d'un fichier

Le fichier à imprimer, créé sur mini-disque par une application CMS, sera transmis dans la file d'attente de la 3812 au moyen de la commande PRT3812, dont la syntaxe est :

```
PRT3812 fn [ft [fm]] [(OPTIONS)]
```

Figure 120. Commande EXEC PRT3812

Les différentes options permettent d'agir sur la présentation du document final; c'est ainsi que l'on peut intervenir sur:

<b>COPIES</b>	Le nombre de copies souhaitées
<b>ORIENT</b>	L'orientation du document au moment de l'impression (on pourra travailler en mode "portrait" (présentation verticale) ou en mode "paysage" (présentation horizontale))
<b>LMAR</b>	La marge de gauche
<b>TMAR</b>	La marge supérieure
<b>FONT</b>	La taille des caractères
<b>LPI</b>	La dimension de l'interligne
<b>PLENGTH</b>	La longueur de la page (11.69" pour format A4)
<b>CC</b>	L'interprétation du premier caractère de chaque ligne comme un caractère de contrôle d'impression (code ASA)

Options et valeurs par défaut (toutes les valeurs nn.nn reprises ci-dessous sont à introduire en "pouces") :

<b>COPIES n</b>	1 par défaut
<b>ORIENT N/W</b>	N ("portrait") par défaut sauf si le fichier à imprimer a pour "filetype" LISTING; dans ce cas, et si l'on n'a pas spécifié l'option FONT, le défaut est W ("paysage").
<b>LMAR nn.nn</b>	Les défauts sont : 0.5 si ORIENT est N et 1 si ORIENT est W
<b>TMAR nn.nn</b>	0.5 par défaut quelle que soit l'orientation
<b>FONT xx</b>	COBE10 si l'on souhaite 10 caractères/pouce
<b>LPI nn.nn</b>	COBE12 si l'on souhaite 12 caractères/pouce
<b>PLENGTH nn.nn</b>	6.0 par défaut (nombre de lignes au pouce)
<b>CC</b>	11.69 par défaut en ORIENT N et 8.27 en ORIENT W est à préciser sauf si le fichier à imprimer a pour "filetype" PRINT ou LISTING.

*Exemple :*

Pour imprimer un fichier TEST LISTING A qui contient des enregistrements de 132 caractères et 60 lignes par page, et pour une impression en mode "paysage" (ORIENT W), en petits caractères (font COBE12) sur un document de format A4 (8.27" x 11.69"), on entrera la commande

```
PRT3812 TEST LISTING A (FONT COBE12 TMAR 0.35  
LMAR 0.35 LPI 8.0 ORIENT W
```

*Remarques :*

- L'option "COPIES n" provoque l'impression de la page 1 en "n" exemplaires, puis celle de la page 2 en "n" exemplaires ... Si l'on ajoute l'option "COLLATED" (ce qui ralentit l'imprimante), on obtient les "n" exemplaires du document triés.

- La surimpression est ignorée.

## 5.10.2 Facturation des coûts

L'utilisation de l'IBM 3812 sera facturée 3 frs la page; les factures, payables par transfert d'un compte Patrimoine ou d'un crédit ordinaire, seront établies mensuellement et adressées aux services utilisateurs. Toute personne désireuse de recourir à cette imprimante fera donc préalablement remplir et signer par son chef de service une autorisation d'accès (voir en annexe); cette autorisation sera liée au "userid" désigné par le demandeur.

## 5.10.3 Modalités d'exploitation

Les fichiers à imprimer sont envoyés, par la commande PRT3812, dans le lecteur virtuel d'une machine spécialisée, qui gère les sorties sur l'imprimante laser en tenant compte des paramètres introduits par l'utilisateur dans la commande PRT3812.

Le document final, présenté sur papier A4, est précédé d'une page de garde sur laquelle sont reprises les informations de distribution.

## 5.11 Utilisation des bandes et des cassettes sous VM/CMS

Les commentaires qui suivent ne se veulent pas exhaustifs: ils constituent une première approche ayant pour seule intention d'aider les utilisateurs à couvrir les principaux usages de bandes magnétiques qu'ils pourraient rencontrer sous CMS et à les mettre en garde contre les erreurs les plus fréquentes. De plus amples informations peuvent être obtenues dans les brochures suivantes:

```
IBM VM/SP: CMS USER'S GUIDE (SC19-6210)
CMS COMMAND AND MACRO REFERENCE (SC19-6209)
```

### 5.11.1 Notions générales

Avant d'entrer plus en détail dans les aspects techniques d'utilisation, il est bon de rappeler quelques notions relatives aux bandes magnétiques.

En premier lieu, il faut distinguer les 2 sortes de supports bandes magnétiques, d'une part les cassettes magnétiques en service depuis mai 89 appelées à se généraliser et d'autre part les bandes magnétiques (2400 pieds) appelées à disparaître. Les cassettes sont utilisées en densité unique 38 K BPI. Les bandes sont utilisées en densité 1600 BPI ou 6250 BPI. C'est cette dernière densité qu'il convient d'adopter en retenant bien qu'une bande ne peut contenir de fichiers **qu'en une seule densité**

Deuxièmement, il faut savoir que l'on peut enregistrer plusieurs fichiers sur une même bande/cassette. Ces différents fichiers seront alors séparés par un enregistrement spécial appelé TAPEMARK (TM). Le dernier fichier d'une bande/cassette sera suivi de deux TM indiquant ainsi la fin de la partie enregistrée de la bande/cassette.

ATTENTION! On peut toujours ajouter des fichiers à la suite des fichiers existants ou ajouter des enregistrements à la FIN du dernier fichier, mais on ne peut JAMAIS écrire dans des fichiers intermédiaires au risque de perdre toutes les informations suivantes (sauf si l'on désire abandonner volontairement ces informations).

Une bande/cassette contenant trois fichiers se présentera donc comme suit:

```
fichier 1 TM fichier 2 TM fichier 3 TM TM
Si l'on y ajoute un quatrième fichier, on aura:
fichier 1 TM fichier 2 TM fichier 3 TM fichier 4 TM TM
Si l'on réécrit le fichier 1, on aura:
fichier 1 TM TM
```

Il est conseillé d'utiliser les bandes/cassettes avec les labels standards d'IBM. Ces labels sont appelés HEADER LABEL (HDR) et TRAILER LABEL (EOF). Les bandes/cassettes comportant ces fichiers HDR et EOF sont appelés STANDARD LABEL (SL) par opposition aux bandes/cassettes NO LABEL (NL) où ils sont absents.

Le fichier HDR est composé de deux enregistrements de 80 caractères dont le premier contient notamment:

```
en position 1 le mot 'hdr1'
en position 5 le nom du fichier principal
en position 22 le nom de la bande/cassette
en position 32 le numéro de séquence
                  du fichier principal
```

Le second contient notamment:

```
en position 1 le mot 'hdr2'
en position 5 le format des enregistrements
                  (f,v,u)
en position 6 la longueur des blocs
en position 11 la longueur des enregistrements
en position 16 la densité utilisée (3 pour
                  1600 BPI, 4 pour 6250 BPI
                  6 pour 38K BPI)
```

Le fichier EOF est identique au fichier HDR à l'exception des mots 'hdr1' et 'hdr2' qui sont remplacés par 'eof1' et 'eof2' ou 'eov1' et 'eov2'.

Remarque:

Le premier fichier HDR d'une bande/cassette contient un troisième enregistrement qui précède les deux enregistrements décrits ci-dessus et qui contient notamment:

```
en position 1   le mot 'voll'  
en position 5   le nom de la bande/cassette
```

Une bande/cassette SL contenant deux fichiers se présentera donc comme suit:

```
HDR TM fichier1 TM EOF TM HDR TM  
fichier2 TM EOF TM TM
```

Afin de minimiser les risques d'erreurs dans l'utilisation des bandes/cassettes magnétiques sous CMS, nous vous recommandons vivement de TOUJOURS UTILISER DES BANDES/CASSETTES STANDARD LABEL: ainsi vous bénéficierez des contrôles effectués par CMS à l'ouverture d'un fichier pour vérifier si la bande/cassette et le fichier renseignés par le programme correspondent bien aux informations stockées dans le label de la bande/cassette accédée (avec des bandes/cassettes NL, ces contrôles n'ont pas lieu).

Signalons enfin que des informations telles que mot-de-passe, date d'expiration, date de création et générations multiples (GDG) qui sont enregistrées dans le label des bandes SL et gérées automatiquement par le système dans une machine MVS sont disponibles uniquement à des fins documentaires en CMS: leur utilité sortant du cadre de cette note, nous invitons les utilisateurs qui veulent les exploiter sous CMS à se référer aux brochures IBM données en référence.

Tous les utilisateurs qui le peuvent ont avantage à ne plus utiliser que des cassettes magnétiques. Ces avantages sont un moindre coût, une plus grande fiabilité, une plus grande capacité et une plus grande performance. La réservation de ces cassettes se fait via le "contrôle des opérations" (tél. 56.49.29) qui se chargera de les initialiser. L'initialisation consiste à attribuer un "volume label", numéro de 6 chiffres, à cette cassette. A tout moment, la commande CATALK7 permet à l'utilisateur d'obtenir la liste des numéros des cassettes de son service ainsi que quelques renseignements quant à leur dernière utilisation en lecture ou écriture. Par souci de sécurité, une vérification a lieu lors du MOUNT de chaque cassette ; elle interdit à un service d'utiliser une cassette affectée à un autre service.

### 5.11.2 Accès aux bandes sous CMS

Au SEGI, les dérouleurs de bandes et cassettes ne sont connectés qu'à certaines machines batch. Il n'est donc pas possible d'utiliser des bandes/cassettes en interactif: seul le système batch offre cette possibilité.

Tous les travaux CMS requérant l'usage de bandes/cassettes magnétiques seront donc envoyés vers la machine batch pour exécution. Pour signifier à cette machine batch que le job demande à disposer de dérouleurs de bandes/cassettes, il faudra coder dans la carte /JOB ou dans la commande SB le paramètre 'TAPES n', 'CAS n' ou 'CAS3490 n' où n indique respectivement le nombre de dérouleurs de bandes, de cassettes de type 3480 ou de cassettes de type 3490 demandés. Ce sera le travail de la machine gestionnaire BATCH de faire exécuter le job dans une machine disposant du nombre de dérouleurs demandé.

En pratique les jobs peuvent utiliser le nombre maximum de bandes et/ou de cassettes associé à la classe dans laquelle ils sont soumis (voir «Tableau des limitations par classe (travaux batch)» à la page 85).

Il faut remarquer que les travaux demandant des dérouleurs de bandes/cassettes peuvent voir leurs délais de passage allongés en fonction de la disponibilité des dérouleurs.

Avant de laisser démarrer un job demandant des dérouleurs, la machine BATCH aura pris soin de connecter les dérouleurs de bandes aux adresses virtuelles 181, 182, 183 ...; les dérouleurs de cassettes 3480 aux adresses virtuelles 288, 289, 28A,...; et les dérouleurs de cassettes 3490 aux adresses virtuelles 28C, 28D, 28E,... Dans le job, il faudra alors utiliser la commande exec MOUNT pour demander à l'opérateur de monter la bande/cassette que vous voulez utiliser. Cette commande a la syntaxe suivante :

```
MOUNT volser [On vaddr] [ NOring ] [ SL ]  
                Ring      [ NL ]
```

Figure 121. Commande EXEC MOUNT

Remarque: volser est le nom de la bande/cassette et vaddr est l'adresse virtuelle du dérouleur.

Par défaut on aura: MOUNT volser ON 181 NORING SL

Seul le paramètre volser est obligatoire.

La commande MOUNT enverra un message à l'opérateur pour qu'il monte la bande/cassette spécifiée par volser et vérifiera si cette bande/cassette correspond bien aux caractéristiques demandées (ring/noring et sl/nl). Si la bande/cassette est SL, la commande MOUNT vérifiera également que la bande/cassette montée est bien la bande/cassette demandée. La commande MOUNT peut donner lieu aux return codes suivants:

- 0 si la bande/cassette est correctement montée et ready.
- 4 si le paramètre volser est omis dans la commande MOUNT.
- 12 si le paramètre vaddr est invalide.
- 16 si le laps de temps accordé au montage est expiré.
- 20 s'il n'y a pas de dérouleur connecté à l'adresse vaddr.

On peut faire monter successivement plusieurs bandes/cassettes sur un même dérouleur en exécutant plusieurs commandes MOUNT avec la même adresse de dérouleur (vaddr).

Dès qu'un job n'a plus besoin d'un dérouleur, il peut le libérer et ainsi le rendre disponible aux autres travaux batch en exécutant la commande CMS

```
RELTAPE vaddr
```

Figure 122. Commande EXEC RELTAPE

Remarque: vaddr est l'adresse du dérouleur; en l'absence de commande RELTAPE, tous les dérouleurs affectés à un job sont libérés dès que ce job se termine.

ATTENTION: un dérouleur libéré par la commande RELTAPE ne peut plus être utilisé par le job pour un MOUNT ultérieur.

## 5.11.3 Utilisation des bandes et cassettes sous VM/CMS

### 5.11.3.1 Utilisation des bandes et cassettes à partir d'un programme d'application

Dans un programme d'application, tout fichier est défini par un ddname. Ainsi on aura notamment en FORTRAN les ddnames FTxxFyyy, en PL/I le ddname sera l'identificateur du fichier spécifié dans le DECLARE ou dans le string code du TITLE et en assembler le ddname est donné dans le paramètre DDNAME de la macro DCB.

Pour signifier à CMS que l'on désire associer un fichier sur bande/cassette SL à l'un de ces ddnames, il faut avant d'exécuter le programme exécuter les deux commandes CMS suivantes

#### Commande LABELDEF

```
LLabeldef DDNAME FID[?fid][(PERM)]
```

Figure 123. Commande CMS LABELDEF

**ddname** est de DDNAME associé au fichier

**fid** est le nom du fichier; si ce nom dépasse 8 caractères ou contient des caractères spéciaux, il faut coder '?' auquel cas CMS vous invitera à entrer le nom du fichier (maximum 17 caractères) directement après avoir entré la commande LABELDEF. Si le fichier est ouvert en INPUT, la procédure d'open vérifiera si le nom donné dans la commande LABELDEF est le même que le nom renseigné dans le fichier HDR sur bande/cassette. Le cas échéant, l'open donne un return code zéro sinon un abend est provoqué (comme en MVS).

**PERM** est une option qui permet de préserver l'effet de la commande LABELDEF tout au long du job pour autant qu'il n'y ait pas d'abend ultérieur. En l'absence de cette option, la définition introduite par LABELDEF est abandonnée après exécution d'un programme écrit dans un autre langage qu'assembler.

#### Commande FILEDEF

```
Filedef ddname TAPn SL m VOLID tnom  
(RECFM a LRECL lllll BLKSIZE bbbbb DISP mod
```

Figure 124. Commande CMS FILEDEF

**ddname** est le ddname associé au fichier.

**TAPn** indique l'adresse virtuelle du dérouleur sur lequel on a fait monter la bande/cassette

```
TAP1 pour l'adresse 181 (bande 2400)  
TAP2 pour l'adresse 182 ( " " )  
TAP3 pour l'adresse 183 ( " " )  
TAP8 pour l'adresse 288 (cassette)  
TAP9 pour l'adresse 289 ( " " )  
TAPA pour l'adresse 28A ( " " )  
TAPC pour l'adresse 28C (cassette 3490)  
TAPD pour l'adresse 28D ( " " )  
TAPE pour l'adresse 28E ( " " )
```

**m** est le numéro de séquence du fichier que l'on veut accéder sur la bande/cassette;

**tnom** est le nom de la bande/cassette (également vérifié par l'open);

**a** indique le format des enregistrements. On codera:

```
RECFM F pour fixed length  
RECFM FB pour fixed blocked  
RECFM V pour variable length  
RECFM VB pour variable blocked  
RECFM U pour undefined  
RECFM VS pour variable length  
RECFM VBS pour variable blocked
```

On peut également adjoindre le caractère 'A' aux options F, FB, V et VB pour indiquer la présence d'un caractère ASA de contrôle d'avancement de papier (exemple FA, FBA, VA ou VBA).

**lllll** indique la longueur maximale des enregistrements

**bbbbbb** indique la longueur maximale des blocs (max 32760).

**DISP MOD** indique que l'on veut ECRIRE les enregistrements suivants dans le fichier renseigné par 'm' à la suite des enregistrements qui s'y trouvent déjà. (DISP MOD est indisponible avec des bandes NL).

Remarque: il n'est pas utile d'indiquer une densité d'enregistrement avec une bande SL car CMS reprendra automatiquement la densité dans laquelle le label a été enregistré.

Voici comment il faudrait faire pour utiliser une bande/cassette NL sous CMS à partir d'un programme d'application. (Nous vous rappelons encore que, DANS LA MESURE DU POSSIBLE, vous avez toujours intérêt à utiliser des bandes SL)

```
Filedef ddname TAPn NL m
(RECFM a LRECL lllll BLKSIZE bbbbbb [DEN den]
```

**Figure 125. Commande CMS FILEDEF**

Remarque:

**ddname** est le ddname associé au fichier  
**TAPn** indique l'adresse virtuelle du dérouleur sur lequel on a fait monter la bande/cassette  
**m** est le numéro de séquence du fichier que l'on veut accéder sur la bande/cassette.  
**a** indique le format des enregistrements.  
**lllll** indique la longueur maximale des enregistrements  
**bbbbbb** indique la longueur maximale des blocs(max. 32760).  
**den** indique la densité d'enregistrement utilisée sur bande seulement

**ATTENTION:** ce paramètre ne doit être spécifié **QUE** si l'on écrit le **PREMIER** fichier sur la bande: dans les autres cas, CMS prendra la densité utilisée pour les fichiers précédents: den vaut 1600 ou 6250 pour une bande. Ne pas le spécifier pour une cassette.

Exemples:

1. Soit le programme **FORTRAN MYPROG** qui crée le fichier 3 sur la cassette SL 999999 en utilisant des enregistrements de format F, de longueur 100 et bloqués à 2000. On utilise le dérouleur connecté en 288 et le ddname FT10F001, ce qui donne:

```
MOUNT 999999 ON 288 SL RING
LA FT10F001 FID MYDATA (PERM
F1 FT10F001 TAP8 SL 3 VOLID 999999(RECFM FB
LRECL 100 BLOCK 2000
MYPROG
```

2. Soit le programme **MYLECT** qui relit ce fichier. Si MYLECT est exécuté dans le même job que MYPROG, il n'est donc pas nécessaire de refaire une commande MOUNT puisque la bande est toujours montée. On entre donc:

```
F1 FT10F001 TAP8 SL 3 VOLID 999999(RECFM FB
LRECL 100 BLOCK 2000
MYSELECT
```

Rappelons qu'il existe une option PERM de la commande FILEDEF qui permet de garder les FILEDEF'S de fichiers après avoir exécuté un programme pour autant que celui-ci ne se termine pas en abend. Cette option PERM permet donc de condenser les 2 séquences ci-dessus en :

```
MOUNT 999999 ON 288 SL RING
LA FT10F001 FID MYDATA (PERM
FI FT10F001 TAP8 SL 3 VOLID 999999(RECFM FB
LRECL 100 BLOCK 2000 PERM
MYPROG
MYSELECT
```

3. Le même exemple appliqué à une bande NO LABEL donnerait:

```
MOUNT 999999 ON 288 NL RING
F1 FT10F001 TAP8 NL 3(RECFM FB LRECL 100
BLOCK 2000 PERM
MYPROG
MYSELECT
```

### 5.11.3.2 Sauver des fichiers CMS sur cassette et les recréer

Pour sauver des fichiers CMS sur cassette, le SEGI met à votre disposition un SPACE MANAGER (cfr help save). Il est certain que d'un point de vue financier ou sécurité, vous avez intérêt à utiliser ce SPACE MANAGER en lieu et place de toute autre technique de sauvetage. Cependant, si vous êtes amenés un jour à devoir expédier une bande/cassette vers une autre exploitation VM/CMS ou encore si vous recevez des fichiers de l'extérieur créés sous VM/CMS via la commande TAPE DUMP (ce qui constitue la procédure la plus répandue), vous serez amenés à utiliser la commande TAPE comme suit (la commande TAPE étant généralement utilisée uniquement avec des bandes/cassettes NL, nous nous limiterons à celles-ci dans cette note).

Pour sauver un fichier CMS sur bande/cassette NL, on procédera comme suit

1. faire monter la bande/cassette au moyen de la commande MOUNT. La bande/cassette est alors positionnée au début.
2. si le fichier que l'on désire créer n'est pas le premier sur la bande/cassette, il faut positionner correctement la bande/cassette en 'sautant' les x premiers fichiers qui précèdent au moyen de la commande

```
TAPE FSF x [(Vaddr)]
```

**Figure 126. Commande CMS TAPE FSF**

Cette commande a pour effet de dérouler la bande/cassette en comptant les TM's. Ce déroulement s'arrêtera dès que l'on aura rencontré 'x' cassette TM c'est-à-dire à l'endroit où devra commencer le fichier x+1.

3. sauver le fichier CMS au moyen de la commande

```
TAPE DUMP fn ft  $\left[ \begin{matrix} \text{fm} \\ \text{A} \end{matrix} \right] ([vaddr][DEN den][BLKsize blksize])$ 
```

**Figure 127. Commande CMS TAPE DUMP**

Remarque:

**fn** est le filename du fichier,  
**ft** est le filetype du fichier,  
**fm** est le filemode letter du fichier (par défaut: A),  
**vaddr** est l'adresse virtuelle du dérouleur (par défaut: 181)  
**den** indique la densité d'écriture sur une **bande** 1600 ou 6250  
**blksize** indique la longueur du bloc (800, 4096, 32K ou 64K) 64K est vivement recommandé.

La commande TAPE DUMP sauve le fichier dans un format propre à CMS et laisse la bande positionnée à la fin du fichier, c'est-à-dire correctement positionnée pour recevoir le fichier suivant. Si l'on veut encore sauver un autre fichier, on répète le point 3 sinon on écrit 2 TM supplémentaires (nous avons vu que le dernier fichier d'une bande doit être suivi de deux TM) en faisant

```
TAPE WTM 2 [(Vaddr)]
```

**Figure 128. Commande CMS TAPE WTM**

Exemple: pour sauver les fichiers MYPROG FT01F001 B et MYPROG MODULE B sur la cassette 999999 qui contient déjà un fichier et qui est connecté à l'adresse 288, on fera:

```
MOUNT 999999 ON 288 NL RING
TAPE FSF 1 (288
TAPE DUMP MYPROG FT01F001 B(288 BLK 64K
TAPE DUMP MYPROG MODULE B(288BLK 64K
TAPE WTM 2 (288
```

A ce moment, la cassette contient deux fichiers dont le second est composé A LA FOIS de MYPROG FT01F001 et de MYPROG MODULE B.

On peut à tout instant repositionner la bande en son début en exécutant la commande

```
TAPE REW [(Vaddr)]
```

**Figure 129. Commande CMS TAPE REW**

Pour recréer sur minidisque un fichier sauvé avec TAPE DUMP, on utilisera la commande TAPE LOAD comme suit:

```
TAPE LOAD fn ft  $\left[ \begin{matrix} \text{fm} \\ \text{A} \end{matrix} \right] \left( \left[ \begin{matrix} \text{EOT} \\ \text{EOF n} \end{matrix} \right] [vaddr] \right)$ 
```

**Figure 130. Commande CMS TAPE LOAD**

Remarque:

**fn** est le filename du fichier recherché (par défaut on a '\*' c'est-à-dire que le filename ne constitue pas un critère de sélection: tous les fichiers rencontrés et satisfaisant aux autres critères sont rechargés).  
**ft** est le filetype du fichier recherché (par défaut on a '\*' c'est-à-dire que le filetype ne constitue pas un critère de sélection: tous les fichiers rencontrés et satisfaisant aux autres critères sont rechargés).  
**fm** est le filemode letter indiquant sur quel minidisque on veut recharger le ou les fichiers (par défaut: A)  
**vaddr** est l'adresse virtuelle du dérouleur (par défaut: 181)  
**EOT** indique que l'on veut rechercher tous les fichiers satisfaisant aux critères de sélection donnés par fm et ft à partir de la position actuelle de la bande/cassette et jusqu'à la fin de la bande/cassette (c'est-à-dire les 2 TM)  
**EOF n** indique que l'on ne veut effectuer la recherche qu'à partir de la position actuelle de la bande/cassette et jusqu'à ce qu'on ait rencontré n TM (par défaut: EOF 1).

Si nous reprenons notre exemple pour voir comment recharger les deux fichiers MYPROG FT01F001 et MYPROG MODULE sur le minidisque A, on peut faire:

```
MOUNT 999999 ON 288 NORING NL
TAPE FSF 1 (288
TAPE LOAD (288
```

Si l'on désire seulement recharger le fichier MYPROG MODULE, on fera:

```
MOUNT 999999 ON 288 NORING NL
TAPE FSF 1 (288
TAPE LOAD MYPROG MODULE (288
```

Remarques:

- il est facile de lister le nom de tous les fichiers sauvés avec la commande TAPE DUMP sur une bande/cassette, en exécutant la commande:

$$\text{TAPE SCAN} \left[ \begin{array}{l} (\text{EOT}) \\ (\text{EOF } n) \end{array} \right] [\text{Vaddr}]$$

**Figure 131. Commande CMS TAPE SCAN**

**EOT** et **EOF** ont la même signification que pour TAPE LOAD (ils déterminent jusqu'où on veut effectuer la recherche). **TAPE SCAN** ne fonctionne qu'avec des bandes/cassettes enregistrées au moyen de TAPE DUMP.

- on peut également positionner une bande au début d'un fichier au moyen de la commande TAPE SCAN en faisant:

$$\text{TAPE SCAN } fn \text{ ft} \left[ \begin{array}{l} (\text{EOT}) \\ (\text{EOF } n) \end{array} \right] [\text{Vaddr}]$$

**Figure 132. Commande CMS TAPE SCAN**

Ainsi, pour recharger le fichier MYPROG MODULE, on peut faire:

```
MOUNT 999999 ON 288 NORING NL
TAPE SCAN MYPROG MODULE (EOT 288
TAPE LOAD MYPROG MODULE(288
```

Ce qui permettra de recharger sur le minidisque A le premier fichier MYPROG MODULE rencontré sur la cassette.

- les fichiers sauvés avec TAPE DUMP ne peuvent être recréés QUE par la commande TAPE LOAD (cette technique ne permet donc pas de sauver des fichiers en CMS pour les recréer plus tard en MVS ou vice versa).
- rappelons également que dans la mesure du possible, il est préférable d'utiliser le SPACE MANAGER VM plutôt que des bandes/cassettes privées pour sauver et recréer des fichiers CMS. **SPACE MANAGER** regroupe en effet les demandes des différents utilisateurs et optimise ainsi au mieux la demande en montages de cassettes et l'occupation de ces cassettes.

### 5.11.4 Précautions à prendre quand on utilise les bandes/cassettes magnétiques en CMS

Nous avons expliqué dans cette note les principales manipulations que l'on peut réaliser sur les bandes/cassettes magnétiques en CMS. Sans être exhaustifs nous nous sommes contentés d'exposer comment réaliser le plus simplement possible ces manipulations afin de vous permettre de vous familiariser plus rapidement à l'utilisation des

bandes/cassettes sous CMS. Vous constaterez en lisant les brochures données en référence qu'il existe sous CMS bien d'autres moyens d'utiliser les bandes magnétiques. Pour éviter autant que possible des manipulations malheureuses, voici quelques points sur lesquels nous attirons votre attention:

1. La commande TAPE FSF ainsi que d'autres commandes que nous n'avons pas décrites ici, permettent de positionner la bande/cassette avec une facilité qui ne reflète guère le danger de destruction qu'elles sous-entendent: si vous positionnez incorrectement la bande/cassette, ou si vous oubliez tout simplement de la positionner, vous perdrez en tout ou en partie son contenu. Exemple: soit la cassette 999999 qui contient 2 fichiers et on veut y ajouter un troisième fichier. Si l'on fait:

```
MOUNT 999999 ON 288 RING NL
TAPE DUMP * * A (288
```

en omettant de faire suivre la commande MOUNT d'une commande TAPE FSF 2, la commande TAPE DUMP écrira à partir du début de la bande en provoquant ainsi la perte des deux fichiers préalablement enregistrés.

2. Pour les bandes type 2400, tous les fichiers doivent être écrits dans la même densité. Pour être à l'abri de cette erreur possible il suffit donc de ne jamais spécifier le paramètre DEN pour les bandes SL et de systématiquement spécifier 6250 BPI pour tout nouvel usage d'une bande NL.
3. Evitez d'utiliser la commande TAPE avec des bandes/cassettes SL car vous éviterez par la même occasion de démolir un fichier HDR ou EOF. En effet, la commande TAPE positionne la bande/cassette et écrit sur celle-ci sans égard pour ces fichiers (contrairement à la commande FILEDEF): toute erreur pourrait donc alors être fatale.
4. Dans les exemples repris dans cette note, seule l'utilisation du TAPE DUMP nécessite de la part de l'utilisateur l'écriture explicite du TM au moyen de la commande TAPE WTM, cependant il faut savoir qu'un abend du système batch VM ou un cancel du job laissera un fichier ouvert en output dans l'état où il se trouve, c'est-à-dire non clôturé par 2 TM. Une lecture ultérieure de ce fichier se terminera alors en erreur car la condition de fin de fichier ne sera pas détectée.
5. La commande MOUNT vérifie si la bande/cassette est SL ou NL et envoie un message à l'opérateur s'il y a contradiction entre ce qu'a spécifié le programmeur dans la commande MOUNT et ce qui existe sur bande/cassette deux situations peuvent alors se présenter:
  - a. soit l'opérateur s'est trompé de bande/cassette et en recevant le message du MOUNT il rectifie la situation en montant la bonne bande/cassette;
  - b. soit le programmeur a mis NL pour une bande/cassette SL ou vice versa, et dans ce cas l'opérateur est obligé de annuler le job.

De toute manière, une fois que la bande/cassette est montée et acceptée par la commande MOUNT, c'est la responsabilité du programmeur de toujours correctement coder NL ou SL dans les commandes FILEDEF au risque de démolir les

informations déjà enregistrées sur la bande/cassette. En effet, CMS ne vérifie pas systématiquement si la bande/cassette est SL ou NL et fait confiance au programmeur.

6. Terminons par une mise en garde qui, bien qu'évidente, mérite d'être faite étant donné les conséquences qu'elle entraîne: dans les commandes TAPE et FILEDEF, faites attention de coder correctement les paramètres vaddr et TAPn, c'est-à-dire l'adresse virtuelle du dérouleur sur lequel est montée la bande/cassette. Dans un job qui utilise plus d'un dérouleur, se tromper d'adresse équivaut à se tromper de bande/cassette, c'est-à-dire très probablement démolir des informations utiles. Remarquons que cette erreur peut être évitée en travaillant avec des bandes/cassettes SL sauf pour la commande TAPE DUMP qui ne vérifie pas le label.

En conclusion, l'on peut conseiller les principes suivants

1. Evitez de sauver vos fichiers CMS avec la commande TAPE, mais utilisez plutôt le **SPACE MANAGER**
2. Pour tous les autres travaux, utilisez des bandes/cassettes SL sans spécifier le paramètre DEN.

## 5.11.5 Support ASCII en CMS

Le code ASCII est supporté via le paramètre 'OPTCD Q' de la commande FILEDEF :

```
FILEDEF ddname TAPn ... (... OPTCD Q
```

**Figure 133. Commande CMS FILEDEF**

Après cette commande, tout record lu sur l'unité TAPn est converti en EBCDIC après lecture. De même, tout record écrit sur cette unité est converti en ASCII avant écriture.

## 5.12 Space Manager VM

Le S.E.G.I. met à la disposition des utilisateurs un système de sauvetage sur cassette magnétique de fichiers CMS appelé SPACE MANAGER.

Ce système comporte:

1. Une commande de sauvetage
2. Une commande de restore
3. Une commande de listage du catalogue des fichiers sauvés
4. Une commande de suppression d'entrées dans ce catalogue
5. Une machine virtuelle déconnectée effectuant les ordres de transfert et de suppression

6. Une commande d'extraction des messages et fichiers renvoyés dans le lecteur virtuel de la machine de l'utilisateur.

### 5.12.1.1 Sauvetage

```
SAVE fn [ft] [(options)]
```

**Figure 134. Commande EXEC SAVE**

Options:

- FROM user[/password] addr
- TO user[/password]
- ERASE
- COMMENT comment

Remarque: fn et ft peuvent contenir des astérisques suivant les mêmes règles que pour la commande LISTFILE.

- L'option **From** permet d'indiquer le propriétaire et l'adresse du minidisque contenant les fichiers à sauver. La spécification du password est nécessaire si le minidisque appartient à une autre machine virtuelle que celle qui effectue la demande de sauvetage. Le password à indiquer sera le password du minidisque en lecture si l'option 'Erase' n'est pas employée; ce sera le password en écriture si l'option 'Erase' a été spécifiée. Par défaut, on a: FROM \* 191 (propriétaire = demandeur). Les restrictions d'accès au minidisque d'un autre utilisateur suivent les règles ACF2 (voir «Gestion et protection des minidisques» à la page 74).
- L'option **To** permet d'indiquer le propriétaire de la copie sur cassette des fichiers sauvés. Ce propriétaire doit nécessairement être connu comme machine virtuelle et le password général de cette machine doit être spécifié si ce propriétaire est différent du demandeur. Par défaut on a: TO \* (propriétaire = demandeur).
- L'option **Erase** permet de demander la suppression des fichiers du minidisque d'origine après le sauvetage sur cassette (=archivage). Cette suppression a lieu uniquement si lors des opérations de copie sur bande, il est possible de prendre en écriture le minidisque d'origine.
- L'option **Comment** permet d'introduire un commentaire associé aux copies sur cassette. Ce commentaire est limité à 33 caractères et doit être la dernière option introduite (tout ce qui suit le mot clé 'Comment' étant pris comme le contenu du commentaire).

Les demandes sont envoyées à la machine virtuelle de transfert qui les traite le soir. Les sauvetages ont lieu sur des cassettes gérées automatiquement. Les copies sur cassettes existent en double exemplaire.

Si l'effacement des fichiers n'a pu avoir lieu par suite de l'impossibilité de prendre le minidisque d'origine en écriture, il appartient à l'utilisateur d'effectuer éventuellement cette opération une fois le transfert réalisé. L'utilisateur reçoit notification du transfert dans le lecteur de sa machine virtuelle.

### 5.12.1.2 Restore

```
RESTORE fn [ft [date [time]]] [(options)]
```

Figure 135. Commande EXEC RESTORE

options:

- TO user[/password] addr
- FROM user[/password]
- IMM

L'astérisque peut être employé dans chacun de ces 4 paramètres pour désigner les fichiers de façon générique suivant la convention habituelle(seul ou comme suffixe du nom tronqué).

- L'option **To** permet d'indiquer le propriétaire et l'adresse du minidisque sur lequel recharger les fichiers. L'adresse indiquée est soit une adresse numérique habituelle soit 'RDR' signifiant que les fichiers doivent être transmis dans le lecteur de la machine virtuelle désignée. Si, au moment des opérations de rechargement sur disque, le minidisque indiqué n'est pas accessible en écriture, les fichiers sont automatiquement envoyés dans le lecteur de la machine spécifiée. La spécification du password est nécessaire si le minidisque appartient à une autre machine virtuelle que celle qui effectue la demande de rechargement. Le password à indiquer est le password du minidisque en écriture. Par défaut, on a: TO \* 191 (propriétaire = demandeur). Les restrictions d'accès au minidisque d'un autre utilisateur suivent les règles ACF2 (voir «Gestion et protection des minidisques» à la page 74).
- L'option **From** permet d'indiquer le propriétaire de la copie sur cassette des fichiers sauvés. Ce propriétaire doit nécessairement être connu comme machine virtuelle et le password général de cette machine doit être spécifié si ce propriétaire est différent du demandeur. Par défaut on a: FROM \* ( propriétaire = demandeur ).
- L'option **Imm** a pour effet de donner une urgence plus importante à l'opération de restore demandée.

Remarque:

Si plusieurs fichiers de même nom répondent aux critères d'identification, c'est le dernier dans l'ordre chronologique de sauvetage qui est retenu pour le rechargement. Le cas échéant, il peut donc être nécessaire de préciser davantage (date, time) le fichier choisi.

Les demandes sont envoyées à la machine virtuelle de transfert et traitées toutes les 3 heures (option IMM) ou le soir (valeur par défaut);

La notification du restore est envoyée dans le lecteur du demandeur, et du récepteur s'il est différent.

### 5.12.1.3 Listage du catalogue

```
LISTSAVE [fn [ft [date [time]]]] [(options)]
```

Figure 136. Commande EXEC LISTSAVE

options:

- TERM ou PRINT ou DISK
- FROM userid
- ORDER

Les paramètres fn, ft, date, time peuvent être omis (valeur par défaut '\*'), il peuvent être spécifiés complètement ou de manière générique (c'est-à-dire tronqués et accolés du caractère '\*' suivant les règles habituelles).

- **Term** indique que le listage doit avoir lieu au terminal; **Disk** indique que le listage doit être enregistré dans le fichier de nom CATALOG SPACEMAN A; **Print** provoque l'envoi du listage sur l'imprimante;
- L'option **Order** provoque la présentation du listage dans l'ordre alphabétique des fichiers (par défaut, c'est l'ordre chronologique);
- L'option **From** permet de spécifier un propriétaire autre que le demandeur (pris par défaut).

**LISTSAVE** sans paramètre donne la totalité du catalogue relatif à l'utilisateur en session.

Contrairement aux autres, cette commande ne provoque pas une action différée de la part de la machine virtuelle déconnectée, mais accède directement au catalogue pour en obtenir les informations souhaitées.

### 5.12.1.4 Suppression d'entrées dans le catalogue

```
DELSAVE fn [ft [date [time]]] [(options)]
```

**Figure 137. Commande EXEC DELSAVE**

options:

- FROM user[/password]
- KEEP nb

Les mêmes conventions que ci-dessus sont applicables aux paramètres fn, ft, date, time.

- L'option **From** permet de désigner un autre propriétaire que le demandeur. Le password à indiquer dans ce cas est le password général de la machine virtuelle correspondante. Par défaut, on a: FROM \*
- L'option **Keep** est utilisée pour restreindre l'effet de l'opération de suppression aux versions les plus anciennes d'un fichier de nom donné.

Par défaut, on a: KEEP 0 signifiant que l'on conserve 0 version des fichiers répondant aux critères d'identification indiqués. KEEP n signifie que l'on conserve les n fichiers de même nom répondant aux critères de sélection dont le sauvetage est le plus récent.

Les demandes sont transmises à la machine virtuelle déconnectée qui les effectue dès qu'elle le peut. La notification est envoyée au demandeur s'il est en session.

Le coût des demandes de suppression est d'autant moins élevé que l'identification des fichiers est précise. Le coût en sera minimal lorsque fn, ft, date et time sont spécifiés complètement.

## 5.12.2 Machine virtuelle de transfert

Cette machine active en permanence reçoit dans son lecteur des ordres de transfert et de suppression d'entrées du catalogue. Elle gère ce catalogue qu'elle est seule à accéder en écriture et réalise toutes les opérations liées au transfert: réservation d'unités à cassettes magnétiques, sélection des cassettes de sauvetage, double copie, mise à jour du catalogue, notifications aux utilisateurs, etc...

### 5.12.2.1 Extraction des messages et fichiers envoyés dans le lecteur

```
SPRECUP [mode]
```

**Figure 138. Commande EXEC SPRECUP**

La machine déconnectée envoie dans le lecteur des machines virtuelles qui lui ont demandé des opérations:

- Des messages,
- Des fichiers dans la forme accessible par DISK LOAD

La commande SPRECUP (Space Manager Recuperation) permet d'extraire du lecteur les éventuels messages envoyés par la machine virtuelle au cours des transferts, ainsi que de recharger les fichiers qui s'y trouvent à la suite de l'exécution de "restore" précédemment demandés. Ceux-ci sont rechargés sur le minidisque de fmode spécifié par le paramètre mode (par défaut 'A').

L'utilisateur faisant un usage fréquent des commandes du Space Manager trouvera indiqué d'introduire cette commande dans un fichier PROFILE EXEC.

### 5.12.2.2 Sauvetage et récupération de fichiers VSAM

Il est possible de sauver sur cassette les fichiers VSAM par la commande:

```
VSAMSAVE fm dsname [( fn ft D)]
```

**Figure 139. Commande EXEC VSAMSAVE**

**fm** désigne le "file-mode" du minidisque OS contenant le fichier VSAM à sauver. Ce "file-mode" aura été défini par la commande CMS ACCESS préalablement à la demande de sauvetage; dsname constitue le nom du fichier VSAM sous la forme de la suite des index successifs séparés par des blancs constituant le nom du fichier VSAM.

**fn ft** désigne le nom sous lequel le fichier est enregistré dans le catalogue des fichiers sauvés. Par défaut, fn ft sont les deux premiers niveaux d'index du fichier VSAM. ft est égal à VSAM si le fichier VSAM n'a qu'un niveau d'index. L'entrée ainsi constituée dans le catalogue possède la même forme que pour les fichiers CMS et peut donc faire l'objet des commandes LISTSAVE, DELSAVE, et RESTORE.

Lors de la demande de sauvetage, le fichier VSAM est immédiatement "unloadé" vers le spool à destination de la machine de transfert qui le recopiera sur cassette en même temps que les fichiers CMS à sauver. Le catalogue contenant le fichier à sauver doit être défini au moyen de la commande

```
DLBL IJSYSCT fm DSN dsname (PERM VSAM)
```

A défaut, l'utilisateur sera invité à entrer cette commande.

Lors d'un "restore" d'un fichier VSAM par la commande RESTORE, la commande SPRECUP fournit les messages comme pour les autres demandes, mais il faut en outre employer la commande

```
VSAMLOAD fm dsname [ ( fn ft D)]
```

**Figure 140. Commande EXEC VSAMLOAD**

pour extraire du reader de la machine utilisateur le fichier et le recharger dans le fichier VSAM de nom "dsname" préalablement défini sur le disque OS accédé en mode "fm". Cette commande VSAMLOAD prend, dans le reader, le fichier de nom fn ft ou, à défaut, le premier fichier de classe V envoyé par la machine de transfert; on retrouve dans le spool l'identification fn ft du fichier telle qu'elle figure dans le catalogue des fichiers sauvés.

## 5.13 Modification des caractéristiques d'une machine virtuelle

Chaque machine virtuelle est identifiée dans le système VM par un certain nombre de paramètres (nom de la machine, mots de passe, espace-disque autorisé, ...) enregistrés dans une sorte de catalogue appelé CP DIRECTORY.

Deux commandes (CMS DIRMAINT et CMS SPACE) sont mises à la disposition des utilisateurs pour leur permettre de modifier eux-mêmes certains de ces paramètres, sans affecter ceux qui gèrent l'utilisation des ressources du système ou qui influencent les performances (aucune interférence n'est donc possible entre les modifications apportées par les utilisateurs et celles introduites par le SEGI).

La commande CMS DIRMAINT affecte les ressources générales tandis que la commande CMS SPACE est spécifique aux manipulations de mini-disques (allocation, suppression, transfert, ...).

### 5.13.1 Commande CMS DIRMAINT

#### 5.13.1.1 Syntaxe générale de la commande

```
DIRMaint opérande
```

**Figure 141. Commande EXEC DIRMAINT**

L' "opérande" indique le type d'opération demandée et DIRMaint peut être abrégé en DIRM

Les différentes opérations et leur syntaxe particulière sont détaillées dans le 'HELP' de la commande (entrer au terminal Help DIRM). C'est pourquoi nous ne donnons ci-dessous qu'un bref aperçu des types d'opérations autorisées.

#### 5.13.1.2 Opérations autorisées

1. *Listage de l'entrée de DIRECTORY relative à la machine de l'utilisateur*

```
DIRM REView
```

**Figure 142. Commande EXEC DIRM REVIEW**

Les informations fournies par cette commande (= paramètres de la machine virtuelle) sont envoyées dans le "reader" de la machine émettrice, où elles peuvent être lues par l'utilisateur.

2. *Réception de messages*

```
DIRM QLog
```

**Figure 143. Commande EXEC DIRM QLOG**

Cette commande envoie dans le "reader" de la machine émettrice les réponses fournies par DIRMAINT suite à une mise à jour de DIRECTORY (effectuée par l'utilisateur ou par le SEGI), dans le cas où elles n'auraient pu être transmises au moment de l'opération (machine de l'utilisateur en état de LOGOFF ou de DISCONNECT). A noter que de tels messages sont également transmis dans le "reader" lors de l'exécution de la première commande DIRM d'une session.

3. **.Modifications du mot de passe d'une machine**

Depuis la mise en service du logiciel de sécurité ACF2 la modification du mot de passe des machines par la commande DIRMAINT PW n'est plus autorisée. On se reportera au chapitre «Gestion des mots de passe d'une machine virtuelle» à la page 73 pour ces opérations qui s'effectuent dans la séquence de LOGON.

4. *Listage des mots de passe des mini-disques*

```
DIRM MDPW ...
```

**Figure 144. Commande EXEC DIRM MDPW**

5. *Changement du mode d'accès ou des mots de passe des minidisques*

```
DIRM MDisk ...
```

**Figure 145. Commande EXEC DIRM MDISK**

L'utilisation de cette commande est soumise à l'autorisation du SEGI vu son impact possible sur la sécurité des données. Des informations complémentaires sont disponibles dans le chapitre «Gestion et protection des minidisques» à la page 74.

#### 6. Accès aux minidisques d'un autre utilisateur

Il est possible d'introduire dans le DIRECTORY un LINK permanent vers un minidisque dont est propriétaire un "USERID", ce qui évite la manipulation de "password" pour accéder au minidisque et donc augmente la sécurité. Cette facilité sera introduite par le SEGI pour autant que le propriétaire du minidisque en fasse la demande au contrôle des opérations. Pour les raisons de sécurité évoquées ci-avant, le lecteur est renvoyé au chapitre «Gestion et protection des minidisques» à la page 74.

#### 7. Contrôle sur d'autres machines virtuelles :

Un utilisateur (= une machine virtuelle) peut agir sur les caractéristiques du DIRECTORY d'autres machines, au moyen de la commande:

```
DIRM FOR userid "commande à exécuter"
```

**Figure 146. Commande EXEC DIRM FOR**

Cette possibilité de contrôle requiert toutefois l'accord préalable du responsable du service concerné et la communication au SEGI des noms ("userids") des machines "dépendantes".

8. **Modification de l'imputation des travaux :** moyennant accord du responsable de service, l'utilisateur peut demander au SEGI d'associer plusieurs comptes de travail à sa machine et ainsi modifier, en cours de session, le compte de travail sur lequel ses travaux sont exécutés. Depuis la mise en service du logiciel de sécurité ACF2, ces modifications "dynamiques" des comptes de travail ne sont plus autorisées par la commande DIRMAINT ACCOUNT, mais bien par la commande CP SET ACCount "compte de travail".

#### 9. .Autres modifications de paramètres

- *Point de retour standard des listings*

```
DIRM DISTrib dest-case
```

**Figure 147. Commande EXEC DIRM DISTRIB**

- *Mémoire disponible lors du LOGON*

```
DIRM STORage ...
```

**Figure 148. Commande EXEC DIRM STORAGE**

La mémoire est cependant limitée à la valeur maximum autorisée par le système.

- *Caractères d'édition de lignes*

```
DIRM Term ...
```

**Figure 149. Commande EXEC DIRM TERM**

- *Certaines options particulières (real time, ISAM, SVCOFF, BMX...)*

```
DIRM OPTion ...
```

**Figure 150. Commande EXEC DIRM OPTION**

### 5.13.1.3 Utilisation de DIRMAINT

Nous attirons l'attention des utilisateurs sur les points suivants:

1. Une commande DIRMAINT n'est acceptée par le système qu'après l'entrée du mot de passe de la machine émettrice
2. Certaines commandes conduisent à une mise à jour directe du DIRECTORY, d'autres à une mise à jour différée; on pourra consulter le 'HELP' pour toute information complémentaire (entrer la commande 'Help DIRM').

## 5.13.2 Commande CMS SPACE

### 5.13.2.1 Syntaxe générale de la commande

```
SPACE opérande
```

**Figure 151. Commande EXEC SPACE OPERANDE**

L' "opérande" indique le type d'opération demandée.

Les différentes opérations et leur syntaxe particulière sont détaillées dans le 'HELP' de la commande (entrer au terminal H SEGI SPACE). C'est pourquoi nous ne donnons ci-dessous qu'un bref aperçu des types d'opérations autorisées.

### 5.13.2.2 Opérations autorisées

#### Réservation d'un mini-disque permanent:

```
SPACE ADDdisk userid cuu cylinders
```

Figure 152. Commande EXEC SPACE ADDDISK

Cette commande attribue à la machine "userid" un nouveau mini-disque permanent de x "cylinders" à l'adresse "cuu" pour autant que l'espace maximal toléré (espace contractuel augmenté de 25 cylindres) ne soit pas dépassé.

Une fois alloué, le mini-disque DEVRA être LINKED puis FORMATTED avant toute autre opération.

#### Libération d'un mini-disque permanent:

```
SPACE DROPdisk userid cuu
```

Figure 153. Commande EXEC SPACE DROPDISK

Cette commande retire à la machine "userid" le mini-disque permanent préalablement alloué à l'adresse "cuu". Le mini-disque que l'on souhaite libérer doit toutefois être FORMATTED et NOT LINKED au moment de la demande.

#### Transfert d'un mini-disque vers une autre machine virtuelle

```
SPACE TRANdisk olduserid oldcuu TO newuserid newcuu
```

Figure 154. Commande EXEC SPACE TRANDISK

Cette commande n'est autorisée que si "olduserid" et "newuserid" appartiennent au même groupe et que le demandeur est "olduserid" ou le responsable du groupe.

#### Changement d'adresse d'un mini-disque:

```
SPACE CHNGdisk userid cuu TO newcuu
```

Figure 155. Commande EXEC SPACE CHNGDISK

Le demandeur peut être le "userid" ou le responsable du groupe.

#### Affichage de l'espace autorisé par contrat et de l'espace occupé:

```
SPACE LIMit userid
```

Figure 156. Commande EXEC SPACE LIMIT

### 5.13.2.3 Utilisation de SPACE

Nous attirons l'attention des utilisateurs sur les points suivants:

1. la réservation (ou la libération) de mini-disques permanents par les utilisateurs sera nécessairement effectuée dans les limites du contrat-disque dont ils dépendent; rappelons à cet égard que l'espace garanti par un contrat-disque peut, sur demande du service, être partagé par plusieurs machines virtuelles dont une sera désignée comme responsable de groupe ("CZAR") et pourra, à ce titre, superviser l'espace total attribué au groupe.
2. pour chaque contrat, le système tolère un dépassement <sup>20</sup> maximal de 25 cylindres pour répondre à des besoins importants mais de courte durée.
3. la possibilité de réserver de nouveaux mini-disques permanents ne supprime en rien l'intérêt des mini-disques temporaires vivement conseillés chaque fois que le besoin d'espace-disque peut se limiter à la durée de la session.

<sup>20</sup> Chaque piste en dépassement sera facturée à un tarif de *dissuasion* (25 fois supérieur au tarif normal).

## 6.0 La sécurité du système et ACF2

### 6.1 Généralités

Le logiciel CA-ACF2, distribué par la société Computer Associates, a été choisi par le SEGI pour différentes raisons dont son adéquation à reprendre les "sécurités" déjà en place sur le système d'exploitation du superordinateur et sa conformité aux normes internationales (normes C2 du NCSC).

Le renforcement des protections CP et CMS par ACF2 a introduit de petites différences dans les procédures d'accès à certaines ressources du système de base (procédure de LOGON et protection des minidisques par exemple); elles sont présentées ci-après de même que de nouvelles possibilités de protection offertes par ce logiciel.

### 6.2 Gestion des mots de passe d'une machine virtuelle

#### 6.2.1 Remarques importantes et structure des mots de passe

- Les mots de passe des "userid", encryptés par le logiciel ACF2, sont INACCESSIBLES même pour le SEGI. La stricte CONFIDENTIALITE de ces derniers est à la base de TOUT le système de protection.
- Tous les utilisateurs doivent effectuer les opérations relatives aux mots de passe dans la séquence de LOGON et non, en cours de session, à l'aide de la commande DIRM PW qui conduirait à introduire dans la DIRECTORY un mot de passe inconnu d'ACF2 (et donc inutilisable par le système d'exploitation).
- Chaque nouveau mot de passe introduit doit se composer de 6 à 8 caractères alphanumériques et être différent des 4 précédents.
- Dix tentatives infructueuses CONSECUTIVES d'introduction du mot de passe conduisent à un verrouillage du "userid" concerné qui ne peut être débloqué que par l'intervention du SEGI. Une telle situation pourra imposer la réintroduction d'un nouveau mot de passe.
- Trois tentatives d'introduction d'un mot de passe sont permises durant la MEME SEQUENCE de LOGON. Les mots de passe erronés sont comptabilisés comme "tentative frauduleuse" et peuvent conduire à un verrouillage tel que décrit ci-dessus.

- La durée de vie d'un mot de passe est de 30 jours, délai après lequel l'utilisateur est invité à en introduire un nouveau.

#### 6.2.2 Séquence de démarrage d'une machine virtuelle

- Si l'on ne souhaite pas modifier son mot de passe :

```
=> VM1 userid
. à introduire sur la "mire VTAM"
<- ACFAC2244R Enter cccccc or cccccc/nnnnnn
where cccccc is your current password and nnnnnn is
your optional new password
-> password
<- ACFVLD137I "userid" last system access at .....
. message supplémentaire qui communique à l'utilisateur
des informations sur le dernier LOGON (heure, date et
nom du terminal utilisé)

... démarrage de la session
```
- Si l'on souhaite modifier son mot de passe :

```
-> VM1 userid
<- ACFAC2244R Enter cccccc or cccccc/nnnnnn
where cccccc is your current password and nnnnnn is
your optional new password
-> oldpassword/newpassword
<- ACFAC2246R Enter new ACF2 password again to verify
-> newpassword
<- ACFVLD137I "userid" last system access at .....
<- ACFVLD129I Password successfully altered

... démarrage de la session
```

#### 6.2.3 Agir sur les mots de passe de machines "dépendantes"

L'utilisation de la commande ACF permet à un utilisateur préalablement autorisé (baptisé LEADER) de modifier le mot de passe ou d'interdire toute activité (opération de "suspend") sur un USERID appartenant à une liste de machines virtuelles "dépendantes".

Une telle possibilité peut être intéressante pour la gestion de machines "étudiants".

## 6.3 Gestion et protection des minidisques

### 6.3.1 Généralités sur la sécurité des minidisques

- Le contrôle des minidisques effectué par ACF2 vient se superposer à la protection standard assurée par le système de base (CP) basée sur les informations contenues dans la DIRECTORY ("password" principalement).
- Le logiciel ACF2 est sollicité chaque fois qu'un utilisateur souhaite accéder à un minidisque dont il n'est pas propriétaire (commande CP LINK ou LINK dans la DIRECTORY, commande CMS ACCESS, ouverture d'un fichier en READ ou WRITE).

Si le demandeur n'a pas été autorisé par une "ACCESS RULE" ACF2, les commandes ou tentatives d'accès sont refusées. ACF2 en informe l'utilisateur par le message:

```
ACFAC7913E "demandeur" VIOLATION LINK READ/WRITE  
"cible".Vadr.VOLUME
```

En cas d'autorisation par ACF2, le contrôle est pris en charge par le CP comme précédemment. Il y a donc superposition des deux systèmes de sécurité.

### 6.3.2 Possibilités offertes par ACF2

- ACF2 permet les mêmes protections que le système de base (CP) en supprimant toute manipulation de "password" d'un minidisque et les contraintes qui y sont liées.
- ACF2 permet de créer des règles d'accès plus sélectives que le CP. Il est en effet possible de limiter l'accès à un minidisque à l'ensemble des utilisateurs d'un même service ou à une liste restreinte d'utilisateurs (appartenant ou non au même service) et ce, sans lier les autorisations d'accès à l'utilisation d'un "password".
- La protection ACF2 peut s'appliquer, de manière sélective, à certains fichiers d'un même minidisque, ce qui est impossible dans le système de base (CP).
- ACF2 introduit un mode de protection supplémentaire sur les fichiers, qui peuvent désormais être protégés en mode EXECUTE. De la sorte, un fichier (EXEC ou MODULE) peut être exécuté sans que son contenu puisse être visualisé (ou modifié) par l'utilisateur.

### 6.3.3 Protection des minidisques et fichiers

- Pour la toute grande majorité des utilisateurs, la protection des minidisques sera assurée, par le SEGI, au moment de l'ouverture des comptes et "userids" selon les souhaits de chaque titulaire. Toute modification ultérieure des protections sera introduite par le SEGI pour autant que le propriétaire des minidisques en ait fait la demande au "Contrôle des opérations".
- Dans certains cas, un utilisateur pourrait être autorisé par le SEGI à modifier lui-même les règles d'accès ACF2 relatives à ses propres minidisques ou à ceux d'une liste de "userids" dont il serait le "gestionnaire". L'utilisateur intéressé est invité à prendre contact avec le "Contrôle des opérations".

## 6.4 Quelques protections supplémentaires permises par ACF2

Les possibilités qui suivent permettent d'accroître le niveau de sécurité dans certains environnements bien précis; elles demandent l'intervention du "contrôle des opérations" du SEGI qui accordera les autorisations nécessaires et, si nécessaire, communiquera aux utilisateurs intéressés les procédures appropriées.

### 6.4.1 Accès à une même machine virtuelle à partir de plusieurs USERIDS

Le logiciel de sécurité permet le LOGON sur certaines machines (machines "cible") en s'identifiant par le USERID et le PASSWORD d'autres machines appartenant au même service utilisateur.

De la sorte et en conservant au "password" son caractère INDIVIDUEL:

- un "userid" peut accéder à sa machine mais aussi à d'autres machines "cible" en ne gérant qu'un seul mot de passe, le SIEN;
- plusieurs "userids" peuvent accéder à une même machine "cible" en introduisant LEUR mot de passe et donc sans connaître celui de la machine "cible".

### 6.4.2 Protection par "terminal physique"

Il est possible de n'autoriser l'accès à une machine virtuelle qu'à partir de certains terminaux dont la localisation géographique est connue du système. Un tel contrôle est envisageable pour les terminaux connectés au réseau SNA du SEGI par des lignes "directes", mais exclut les terminaux qui communiquent avec nos installations via des réseaux intermédiaires (INTERNET, PBX, DCS, commuté RTT, ...)

## 6.5 *Quelques messages ACF2*

- <- ACFAC5218E Incorrect ACF2 password entered
  - . Signale, en phase de LOGON, l'introduction d'un mauvais mot de passe; il est suivi par une nouvelle invitation (message ACFAC2244R)
- <- ACFAC2230W ACF2 session terminated
  - . Indique une interruption de la séquence de LOGON; il est précédé d'un message indiquant la raison de l'interruption.
- <- ACFAC2678E New password must be different from old password -- logon failed
  - . Non respect d'une des contraintes de modification des mots de passe.

- <- ACFVLD134W Your password will expire on .....
  - . Information de la date à laquelle le mot de passe ne sera plus valide.
- <- ACFVLD013E Logonid "userid" suspended because of password violation
  - . Blocage du "userid" suite à un nombre trop important d'essais infructueux consécutifs.
- <- ACFVLD017E Password for Logonid "userid" has expired
  - . Information directement suivie du message ACFAC5684W
- <- ACFAC5684W ACF2 Logon password has expired
  - . Se présente au premier LOGON ultérieur à la date d'expiration; il est suivi par une invitation à introduire un nouveau mot de passe.



## 7.0 La communication au SEGI

Remarque préliminaire : le SEGI développe depuis quelques années une infrastructure d'interconnexion des réseaux locaux appelée Interréseau ULg. Cet interréseau déborde largement du cadre VM, puisqu'il concerne aussi les systèmes Unix, les PC et les MacIntosh. Pour plus d'informations, et notamment pour les commandes ou services qui ne seraient pas repris dans le présent guide, voir le "Guide d'Introduction à l'Interréseau ULg", SEGI, zz01-0117.

### 7.1 Introduction

Le SEGI offre une grande variété de services de type "réseau".

Ces services s'appuient, dans l'environnement VM, sur deux systèmes de gestion de communication :

- VTAM/SNA
- TCP/IP

VTAM/SNA constitue le support standard de la communication dans un monde homogène IBM. Il intervient donc dans la communication entre le superordinateur et les terminaux IBM ou assimilés (émulation de terminaux IBM par des microordinateurs par exemple), entre le superordinateur et d'autres ordinateurs IBM (ou non-IBM capables de mettre en oeuvre des protocoles de communication IBM), ainsi que comme passerelle vers les réseaux X25 (tels que DCS pour la Belgique).

TCP/IP offre un ensemble de logiciels de communication indépendants du constructeur, permettant la communication hétérogène avec des ordinateurs de tous types, et en particulier avec le monde UNIX où ce support est très répandu.

Les principaux services "réseau" disponibles au SEGI sous VTAM/SNA sont :

- l'accès interactif au VM à partir de terminaux IBM ou assimilés (terminaux "pleine page" de type 3270), généralement connectés au SEGI par des liaisons permanentes.
- ce même accès interactif offert aux microordinateurs simulant les terminaux 3270 grâce à la combinaison d'un programme sur le microordinateur et de moyens matériels particuliers mis en oeuvre par le SEGI. Par extension, l'échange de fichiers entre le microordinateur et le VM est également possible dans le cadre d'une session interactive. Ces connexions sont possibles sur liaisons fixes, commutées sur le réseau PBX de l'Université.
- l'accès interactif au SEGI au travers des réseaux X25 (DCS), et, dans l'autre sens, la communication offerte à tous les terminaux de l'ULg vers les serveurs accessibles par les réseaux X25.

- le service "RSCS", permettant principalement l'utilisation des imprimantes décentralisées.

Les services "réseaux" offerts sur le VM du SEGI sous TCP/IP doivent principalement être vus comme une communication d'égal à égal entre le VM et un autre système disposant des protocoles TCP/IP. Cet autre système, s'il est localisé à l'ULg, sera généralement intégré dans un réseau local lui-même connecté à "l'interréseau" ULg, qui interconnecte progressivement l'ensemble des réseaux locaux de l'ULg. Dans le cas d'un système extérieur à l'Université, ce système communiquera généralement avec les systèmes de l'ULg et avec le VM en particulier au travers de l'Internet dont la couverture est mondiale et dans lequel l'interréseau de l'ULg est intégré.

Les services disponibles dans ce cadre sont donc généralement symétriques et les plus importants sont TELNET et sa variante "pleine page" TN3270 (accès interactif), FTP (échange de fichiers), SMTP (courrier électronique).

Les services les plus courants sont décrits dans les chapitres qui suivent. Certains services font l'objet dans ce guide d'une simple introduction dans la mesure où une publication spécifique existe et est disponible au bureau de documentation du SEGI (Utilisation des microordinateurs au travers des équipements 7171, utilisation du réseau DCS), ou dans la mesure où l'information "on-line" est suffisante. Une description des moyens matériels de type "réseau" implantés par le SEGI figure également ci-après.

### 7.2 Réseau Interactif VTAM/SNA

En dehors de la communication TCP/IP, les terminaux et microordinateurs de l'Université accèdent donc au VM du SEGI sous le contrôle de VTAM/SNA. Lors de l'établissement de la connexion (mise sous tension pour un simple terminal, activation d'un programme de communication sur un microordinateur, éventuellement complétée par les opérations liées au passage au travers du réseau physique PBX, et s'il y a lieu, au travers du réseau téléphonique commuté), l'écran affiche la mire VTAM sur laquelle sont indiquées les principales destinations accessibles, parmi lesquelles :

1. Le VM du SEGI, sur lequel l'utilisateur entre en session en entrant VM1 suivi du userid. Il est alors mis en communication avec le système VM et est invité à entrer son mot de passe. Au LOGOFF ou au DISCONNECT, il revient automatiquement au menu initial et peut éventuellement choisir une autre destination.
2. Le service VMDCS d'accès au réseau X25 de la RTT réservé aux terminaux 3270 réels ou simulés que l'on active en entrant simplement DCS. La suite du dialogue est décrite dans une notice d'utilisation spécifique à ce service : "Accès au réseau X25 par DCS", SEGI, zz01-0105.
3. Le service VMDCS 7171 d'accès au réseau X25 de la RTT, réservé aux micros et terminaux asynchrones en pur mode ASCII transparent. Ce service est activé en

entrant DCS7171. La suite du dialogue est décrite dans la notice dont la référence est donnée ci-dessus.

D'autres destinations, ne figurant pas sur le menu VTAM, sont également possible au travers du réseau VTAM/SNA, par exemple vers les autres Universités belges équipées en IBM. L'utilisateur intéressé est prié de se mettre en rapport avec le SEGI pour les procédures de mise en connexion dans ces cas particuliers. Les utilisateurs d'autres institutions ou organismes connectés au SEGI devront généralement suivre les procédures définies au niveau de leur propre centre de calcul pour atteindre le SEGI.

L'accès interactif au VM du SEGI est également possible au travers des réseaux X25 : DCS (numéro d'appel du SEGI : 2410408, précédé de 0206 en international).

## 7.3 Réseau networking

Le réseau RSCS, du nom du logiciel (VM/RSCS) fonctionnant sous VM, assure l'ensemble des fonctions de transfert de fichiers ou de messages disponibles sur le VM du SEGI (à l'exclusion des fonctions similaires offertes dans le cadre TCP/IP). Chaque système d'exploitation constitue un NOEUD du réseau RSCS qui donne accès à un certain nombre de COMPOSANTS qui sont, soit les "userid" connus des différents NOEUDS du réseau, soit les lecteurs et imprimantes virtuels ou réels connectés sur l'un ou l'autre de ces NOEUDS du réseau.

Les possibilités offertes sont:

1. Echange de messages entre "userid" VM connus des NOEUDS du réseau (cfr. commande TELL « Echange de messages entre machines virtuelles: TELL» à la page 36).
2. Echange de fichiers entre "userid" VM connus des NOEUDS du réseau (cfr. commande SENDFILE «Transmission de fichiers.» à la page 37).
3. Possibilité d'envoyer un job batch vers le superordinateur du SEGI, quel que soit le terminal utilisé pour la transmission. De même, il est possible de demander l'impression d'un fichier (produit ou non par un job batch) sur une quelconque des imprimantes connectées au NOEUD SEGI, ou de récupérer ces fichiers en CMS quelle que soit leur provenance (cfr. ci-après: opérations autorisées).
4. Contrôle du réseau VM (cfr. ci-après: opérations autorisées).

### 7.3.1 Composants du réseau RSCS

Chaque composant accessible via le réseau RSCS s'est vu attribuer un nom. Voici la liste des composants accessibles sur le noeud BLIULG11 du superordinateur du SEGI:

Composants	Noms
Imprimante salle des machines SEGI	SYSTEM
Imprimantes <i>remotes</i>	PRLn <sup>21</sup>
Lecteur virtuel d'une machine VM	userid

### 7.3.2 Opérations autorisées

#### 7.3.2.1 Généralités

Les noms des NOEUDS et des COMPOSANTS sont utilisés pour signifier aux logiciels du réseau RSCS la destination que l'on veut donner à un job ou à un fichier de sortie. A cet effet, on indiquera le noeud et le composant de ce noeud au moyen de la commande CP TAG (voir détails plus loin).

Par défaut, les fichiers d'impression d'un job VM sont envoyés dans le lecteur virtuel de la machine qui en est propriétaire.

#### 7.3.2.2 Soumettre un job VM vers un autre système VM

Pour soumettre un job VM sur un système cible à partir d'une machine virtuelle d'un autre système, on doit envoyer le job dans le reader de la machine RSCS et indiquer à RSCS (au moyen d'une commande TAG) la destination à lui donner, c'est-à-dire le noeud et le reader de la machine BATCH située sur le système cible. Pour soumettre un job au superordinateur du SEGI, on écrira donc :

```
SPool PUnch RSCS NOCont NOHold
TAG DEV PUnch BLIULG11 BATCH
PUNCH du fichier contenant le job
```

#### 7.3.2.3 Modifier la destination par défaut des listings produits en VM

En VM, la destination par défaut des listings (imprimante locale VM) peut être modifiée en utilisant la commande TAG et en envoyant les listings dans le reader de la machine RSCS qui, elle, les orientera vers la destination indiquée par TAG.

Par exemple, pour envoyer un listing vers une imprimante PRLx VM, on écrira:

```
CP SPOOL Printer RSCS NOCont NOHold
CP TAG DEV Print PRLx
PRINT fn ft fm
```

<sup>21</sup> Il s'agit d'imprimantes "remote" installées dans certains sites universitaires. *n* désigne un n° attribué par le SEGI pour désigner une imprimante déterminée.

## 7.3.3 Contrôle du réseau networking

### 7.3.3.1 Généralités

Les utilisateurs peuvent intervenir sur les fichiers pris en charge par RSCS pour autant qu'ils en soient propriétaires; les commandes "user" autorisées sont décrites ci-dessous.

Par ailleurs, le responsable d'un "cluster" équipé d'une imprimante PRL.. pourra avoir accès a des commandes privilégiées qui lui permettront de gérer les fichiers en attente d'impression sur cette imprimante (modification des priorités, sortie de documents spéciaux, ...) et d'intervenir lors de certains incidents qui pourraient survenir en cours d'impression (arrêt et redémarrage de l'imprimante). Les responsables de "cluster" intéressés par ce contrôle sont invités à prendre contact avec le "contrôle des opérations" pour déterminer les modalités d'implantation d'une telle solution.

### 7.3.3.2 Commandes "user"

La syntaxe générale de la commande est la suivante:

SMMsg RSCS opérande

Figure 157. Commande CP SMSG

L' "opérande" indique le type d'opération demandée.

1. Modification des caractéristiques d'un fichier qui est en attente d'impression sur une imprimante:

```
SM RSCS CHange { *
                spoolid }
[PRIority nn][CLass c][COpy nnn][DIst distcode]
[FOrm ccccccc][PRIority nn][CLass c][COpy nnn]
[DIst distcode][FOrm ccccccc]
```

Figure 158. Commande CP SM RSCS

#### Remarque:

\* est le numéro d'origine que le fichier a obtenu dans le "spool" au moment de l'émission,  
**spoolid** est le numéro courant que le fichier a obtenu dans le "spool" du noeud destinataire,  
**PRIority** désigne la nouvelle priorité du fichier dans la file d'attente,  
**CLass** désigne la nouvelle classe du fichier,  
**COpy** modifie le nombre de copies à imprimer,

**DIst** modifie le point de retour associé au fichier,  
**FOrm** permet de demander l'impression sur un papier non-standard.

La majorité des paramètres repris dans cette commande ne sont à modifier qu'en fonction des procédures particulières d'exploitation (si elles existent) établies par le responsable du "cluster" sur lequel est connectée l'imprimante.

2. Envoi d'une commande à un autre noeud RSCS

Cette commande permet d'agir sur un fichier qui, après création, est pris en charge par un NOEUD différent de celui sur lequel l'utilisateur est connecté.

SM RSCS CMD nodeid [texte de la commande]

Figure 159. Commande CP SM RSCS

**nodeid** est le nom du NOEUD que l'on souhaite interroger,  
**texte** est la commande qui doit respecter le format demandé par le NOEUD interrogé.

3. Interruption de l'impression d'un fichier

SM RSCS Flush [\*]spoolid [ALL | Hold]

Figure 160. Commande CP SM RSCS

**\*spoolid** est le numéro d'origine que le fichier a obtenu dans le "spool" au moment de l'émission,  
**spoolid** est le numéro courant que le fichier a obtenu dans le "spool" du noeud destinataire,  
**ALL** demande le "cancel" de toutes les copies du fichier concerné,  
**Hold** permet d'interrompre l'impression en cours en plaçant le fichier en état de "hold" (une commande CHANGE permettra de remettre le fichier en attente d'impression).

4. Cancel d'un fichier en attente d'impression

SM RSCS PURge [\*]spoolid

Figure 161. Commande CP SM RSCS

**\*spoolid** est le numéro d'origine que le fichier a obtenu dans le "spool" au moment de l'émission,  
**spoolid** est le numéro courant que le fichier a obtenu dans le "spool" du noeud destinataire,

## 5. Obtenir l'état des fichiers transmis

```
SM RSCS Query File *
```

Figure 162. Commande CP SM RSCS

**File \*** permet d'obtenir l'état des fichiers qui sont sous contrôle du NOEUD local (celui sur lequel on est connecté).

Si les fichiers se trouvent sous le contrôle d'un noeud "remote", on utilisera la commande CMD pour interroger ce dernier (SM RSCS CMD nodeid Q F \*).

**Remarque:** le spoolid d'origine qui peut être utilisé dans les commandes apparaît entre parenthèses dans la réponse à une commande de QUERY.

## 6. Modifier la destination d'un fichier

```
SM RSCS TRANSfer [*]spoolid TO nodeid [userid]
```

Figure 163. Commande SM RSCS

**\*spoolid** est le numéro d'origine que le fichier a obtenu dans le "spool" au moment de l'émission,  
**spoolid** est le numéro courant que le fichier a obtenu dans le "spool" du noeud destinataire,  
**nodeid** est le nouveau NOEUD cible,  
**userid** est le "userid" destinataire (SYSTEM est pris par défaut si ce paramètre est omis).

## 7.4 Le courrier électronique

Le courrier électronique sur le VM, tel que décrit ci-dessous, devrait être abandonné au profit des techniques de type client/serveur mettant en oeuvre un programme client sur PC, Mac ou Unix et un serveur sur Unix.

Les modalités ainsi que les informations nécessaires pour utiliser ces techniques sont disponibles sur le WEB, à l'adresse <http://www.ulg.ac.be/segi/internet/email> où elles peuvent être consultées au moyen d'un programme client tel que Netscape.

Le "courrier électronique" désigne la possibilité offerte aux personnes disposant d'un accès (concrètement un terminal) à un système informatique d'effectuer des échanges d'information avec des utilisateurs du même système ou de tout autre système accessible du premier via des liaisons entre ordinateurs.

Au SEGI, ce type d'échanges est possible:

- à l'intérieur du système VM entre utilisateurs de mêmes systèmes;
- vers des milliers d'ordinateurs dans le monde grâce à l'intégration du SEGI dans les réseaux de recherche internationaux.

Les échanges d'informations possibles de cette manière peuvent être plus ou moins importants. Ils peuvent:

- se limiter à de brefs **messages** vers un autre utilisateur;
- prendre la forme d'une **note** correspondant à la lettre que l'on envoie par le courrier postal traditionnel;
- être constitués de **fichiers** transférés d'un système à l'autre.

### 7.4.1 Courrier électronique local

Des utilisateurs connectés au même système peuvent:

- échanger des messages par les commandes **MSG** (revoir «Facilités particulières du CP» à la page 14) et **TELL** (revoir «Dialogues entre machines virtuelles.» à la page 36), pour autant que les deux partenaires soient simultanément en session;
- s'envoyer des notes (commandes **POST** et **MAILBOOK**).
- s'envoyer des fichiers (commande **SENDFILE**), qui aboutissent dans le "reader" du destinataire qui peut les extraire par la commande **RECEIVE** (revoir «Dialogues entre machines virtuelles.» à la page 36).

Certaines des commandes citées ci-dessus sont décrites dans le chapitre «Dialogues entre machines virtuelles.» à la page 36; les autres sont introduites ci-après.

On y ajoutera la commande **LNAME** qui permet de définir des pseudonymes aux userids habituellement utilisés en VM; cela permet une identification plus simple et facilite la rédaction de notes par des fonctions telles que le rappel automatique des informations d'adresse postale.

### 7.4.2 Le courrier électronique avec l'Internet et les autres réseaux

Le SEGI est intégré dans l'INTERNET mondial et peut donc échanger du courrier électronique avec tout système connecté à ce réseau ou à tout autre réseau disposant d'une passerelle vers l'Internet. Dans ces cas, l'adresse électronique sera plus ou moins complexe en fonction de la destination finale, le nom du noeud étant plus souvent du type

```
systeme.institution.domaine.pays
```

La plupart du temps, il suffira de reprendre l'adresse communiquée par son interlocuteur ou figurant dans l'en-tête d'une note reçue de celui-ci en travers du réseau.

Les commandes à utiliser dans le cadre du courrier électronique sont les suivantes:

1. "**LNAME**", pour enregistrer les indications d'adressage relatives à un partenaire sur le réseau Il est important de **définir d'abord sa propre identification**, car c'est elle qui sera automatiquement ajoutée aux notes envoyées par la suite, comme identité de l'expéditeur. Un utilisateur du SEGI aura une adresse du type "userid AT vm1.ulg.ac.be".

Pour utiliser cette commande, entrer simplement 'LNAME [nickname]' où nickname désigne le pseudonyme que l'on désire associer à l'adresse électronique. Cette commande provoque l'affichage d'un menu, qu'il suffit de compléter et d'enregistrer grâce aux touches fonctions, comme indiqué sur l'écran.

2. "**POST**" pour la préparation et l'envoi de notes, ainsi que pour l'examen du courrier reçu.

Pour la préparation/expédition de notes, cette commande s'écrit:

```
POST { userid AT node } [(options)]
      { nickname }
```

**Figure 164. Commande EXEC POST**

les options principales concernent l'accusé de réception, le "logging" des notes, l'intégration de fichiers dans la note, ...

Cette commande provoque l'affichage d'un écran que l'on complète avec ses données et que l'on contrôle par les touches fonctions.

Pour connaître la syntaxe complète, entrer

### HELP POST

Dans le cas d'une destination interne à EARN/BITNET, "node" ne peut dépasser 8 caractères, mais POST accepte les adresses complexes nécessaires pour adresser les autres réseaux.

**POST** entré au clavier sans paramètres, extrait du "reader" le courrier reçu et permet de l'examiner, répondre, le réexpédier à un autre utilisateur, l'enregistrer dans un fichier "notebook" pour archivage et consultation ultérieure, ... (tout cela par l'utilisation des touches fonctions et du positionnement du curseur).

3. "**MAILBOOK**" qui permet de gérer le courrier archivé dans un fichier "notebook" (lecture, réponse, ...). Cette commande s'écrit:

**MAILBOOK [filename]**

Il suffit alors de se laisser guider par les informations sur l'écran et d'utiliser le positionnement du curseur et les touches fonctions comme indiqué.

A chaque fichier "notebook", est associé un fichier "noteindx", permettant d'optimiser les accès aux notes enregistrées dans le notebook.

## 7.5 TCP/IP

### 7.5.1 Accès au VM du SEGI

Le VM du SEGI peut être accédé à partir de tout site disposant de services TCP/IP, intégré dans l'interréseau ULg ou dans l'Internet. L'adresse TCP/IP à spécifier pour atteindre le VM du SEGI est :

VM1.ULG.AC.BE ou 139.165.32.1

L'utilisation de cette possibilité est soumise aux fonctionnalités TCP/IP offertes sur le système d'origine et aux procédures définies localement.

Les applications TCP/IP sur le VM permettent principalement :

1. d'entrer en session interactive sur le VM du SEGI à partir d'un site éloigné via la commande **TELNET** (ou **TN3270** pour travailler en mode "pleine page")
2. de transférer des fichiers entre le SEGI et un site éloigné via la commande **FTP**.

### 7.5.2 Accès à partir du VM du SEGI

Tout utilisateur en session sur le VM dispose de services TCP/IP de type "client" lui permettant d'entrer en liaison avec tout site TCP/IP de l'Internet mondial ou sur l'interréseau de l'ULg.

Il peut ainsi entrer en session interactive sur de tels systèmes par la commande TELNET. A noter que sa session sera de type "pleine page 3270" uniquement si le système appelé est également de type IBM. Généralement, sa session sera plutôt de type "ligne à ligne". Il peut également entrer en session d'échange de fichier avec un autre système par la commande FTP. Pour obtenir plus d'informations sur l'usage de ces commandes (et sur quelques autres), il suffit d'entrer la commande **HELP TCPIP** et de parcourir les menus d'aide y associés.

## 7.6 Moyens matériels

Les logiciels employés pour la communication au SEGI et vers l'extérieur utilisent évidemment une infrastructure matérielle complexe. Ce chapitre a pour but de décrire les moyens matériels traditionnellement associés à l'environnement VM et mis en oeuvre par le SEGI avant la réalisation de l'interréseau ULg (Voir Guide d'Introduction à l'Interréseau ULg, SEGI, zz01-0117).

## 7.6.1 Les frontaux de communication

Les frontaux de communication sont des équipements qui se connectent à un ou à plusieurs canaux d'ordinateurs et qui gèrent un ensemble de lignes de communication.

Suivant le type de frontal de communication, les fonctions offertes peuvent prendre différentes formes, et concerner différents modes de liaison :

1. Les contrôleurs de type IBM 3745, assurent la gestion de lignes téléphoniques permanentes (lignes louées) ou de lignes téléphoniques commutées, par lesquelles la communication est assurée vers les bâtiments ou les sites éloignés. Au SEGI un contrôleur de ce type gère près de 40 lignes auxquelles sont raccordés des terminaux, des concentrateurs (ou contrôleurs) de terminaux, d'autres ordinateurs, ainsi que des accès au réseau DCS de la RTT. IL assure également l'interface avec le réseau local token ring du SEGI.
2. Les contrôleurs de type 3270 assurent la connexion des écrans et imprimantes de la famille 3270 au travers de câbles coaxiaux de portée limitée à près de 3 km du contrôleur. Au SEGI deux de ces contrôleurs sont connectés directement au canal de l'ordinateur central et permettent de relier jusqu'à 64 terminaux. A noter que ces contrôleurs existent également en version "remote", c'est-à-dire placés sur une ligne téléphonique permanente où ils remplissent le rôle de "contrôleurs de grappe" pour les terminaux de la famille 3270 largement répandus à l'Université.
3. Les équipements de type IBM 7171 installés au SEGI offrent un support particulier destiné aux terminaux "TELETYPE", mais surtout aux microordinateurs, pour permettre à ces équipements de bénéficier du mode de fonctionnement "fullscreen" des terminaux de type 3270. Cette possibilité est réalisée par la simulation au niveau du frontal des fonctions du 3270, pour une grande variété de terminaux "TELETYPE" et de microordinateurs. En ce qui concerne les microordinateurs, une fonction complémentaire de transfert de fichiers entre le microordinateur et le système VM est réalisée par des programmes fournis par le SEGI. La description exhaustive de ces possibilités fait l'objet de notices distinctes disponibles au bureau de documentation du SEGI.
4. L'interface entre le superordinateur et les réseaux locaux sous TCP/IP est assurée par des frontaux spécifiques (IBM 3172).

## 7.6.2 Les concentrateurs

Par concentrateur, on désigne les équipements qui permettent le partage d'une ligne par un ensemble de terminaux habituellement situés dans un aire géographique restreinte (en général le bâtiment ou partie d'un bâtiment). A l'Université, ce type de concentration correspond aux contrôleurs de type 3270 cités plus haut, qui gèrent en général jusqu'à

16 terminaux reliés par des câbles coaxiaux. Cette fonction peut également être assurée par un "gateway" dans des configurations de PC interconnectés entre eux par un réseau local.

Le PBX dont la description fait l'objet du chapitre ci-après, assure également des fonctions de concentration.

## 7.6.3 Concentrateur PBX

### 7.6.3.1 Possibilités offertes par le matériel

Le PBX est une machine à laquelle peuvent se connecter la majorité des équipements asynchrones utilisables en transmission de données (terminaux, microordinateurs, ordinateurs IBM ou non IBM).

A la manière d'un central téléphonique, il permet à un utilisateur qui en fait la demande (porte appelante) d'établir une liaison ASYNCHRONE avec un destinataire (porte appelée) dont il devra introduire l'identificateur. Outre cette fonction de "**COMMUTATION**", il joue le rôle de "**CONCENTRATEUR**" dans la mesure où une liaison peut être demandée, non plus vers une porte de sortie spécifique, mais vers un "**SERVICE**" qui regroupe plusieurs sorties de mêmes caractéristiques. Dans ce cas, les portes affectées au "**SERVICE**" seront partagées entre les différents émetteurs dès l'instant où ces derniers en font la demande.

Un réseau de trois concentrateurs de ce type est actuellement opérationnel (SART TILMAN, VAL BENOIT et BATIMENT CENTRAL). Ces PBX sont reliés "en triangle" au moyen de liaisons haute vitesse (64 Kbps) qui supportent chacune l'établissement de 32 liaisons simultanées sur lesquelles la vitesse de transmission s'effectue à 9600 bps. L'infrastructure en place permet donc aux équipements ASYNCHRONE connectés aux différents PBX d'atteindre les portes du SART TILMAN affectées aux "services" donnant accès aux contrôleurs de communications 7171 du SEGI et ainsi de travailler en émulation 3270 à une vitesse "confortable".

L'installation des PBX à proximité des centraux téléphoniques permet de réaliser toutes les connexions en utilisant le réseau de câbles téléphoniques en place à l'intérieur des sites Universitaires, et donc sans contrainte RTT et à coûts réduits.

### 7.6.3.2 Comment utiliser le PBX ?

L'utilisation du PBX se fait par dialogue avec le "demandeur" d'un SERVICE. Dès le "POWER ON" du terminal, la liaison est autorisée entre celui-ci et le PBX.

**Demande de liaison :**

```

ENTER (au terminal)
RESEAU PBX - SEGI/ULG
SERVICE DEMANDE (ou AIDE/INFO) ?
IBM00 (introduction du service demandé)
CONNEXION SUR LES ORDINATEURS DU SEGI EN
                                EMULATION 3270

CONNEXION REALISEE
ENTER TERMINAL TYPE:
...
dialogue avec la station appelée et
session de l'utilisateur
...
LOGOFF
*****
* DECONNEXION PBX (ST) - ENFONCER 'RETURN' *
*           POUR UNE NOUVELLE CONNEXION *
*****
ENTER (au terminal) pour nouvelle demande

```

N.B.:

- L'utilisateur dispose de 20" pour entrer le SERVICE souhaité; passé ce delai, le PBX annule la demande et envoie les messages:

```

*****
* ATTENTE TROP LONGUE *
* DECONNEXION PBX (ST) - ENFONCER 'RETURN' *
*           POUR UNE NOUVELLE CONNEXION *
*****

```

- Pour éviter le blocage d'une connexion vers le SEGI en l'absence de transfert de données en provenance ou à destination du terminal, le PBX force la déconnexion de la liaison dès que cette dernière reste au "repos" durant 20 minutes; l'utilisateur recevra alors le message de déconnexion Il faut noter que, dans ce cas, la session de l'utilisateur qui a demandé le SERVICE IBM00 se retrouvera en mode DISCONNECTED.

## Mise en attente d'un utilisateur

```
ENTER (au terminal)
RESEAU PBX - SEGI/ULG
SERVICE DEMANDE (OU AIDE/INFO)? :
IBM00 (introduction du service demandé)
IBM00 OCCUPE
xxx UTILISATEUR(S) EN ATTENTE
SOUHAITEZ-VOUS ATTENDRE?
OUI (réponse entrée au terminal)
Si une porte est disponible dans les 5 min.,
l'utilisateur reçoit la séquence:
PRET?
OUI (réponse entrée au terminal)
CONNEXION REALISEE
...
...
Si aucune porte n'est libérée dans les 5
min., la séquence invitant l'utilisateur à
attendre est à nouveau transmise (toutes les
5 min.) et ce, jusqu'à ce qu'il réponde NON.
```

## Services disponibles

Les "services" actuellement disponibles sur le PBX sont les suivants:

<b>AIDE</b>	pour obtenir la liste des services utilisables
<b>IBM00</b>	pour accéder aux systèmes d'exploitation du SEGI en émulation 3270 (noeud BLIULG11)
<b>IBM10</b>	pour accéder aux systèmes d'exploitation du SEGI en mode "ligne à ligne" à la vitesse de 1200 bps (noeud BLIULG11)
<b>IBM11</b>	pour accéder aux systèmes d'exploitation du SEGI en mode "ligne à ligne" à la vitesse de 300 bps (noeud BLIULG11)

## 7.6.4 L'interréseau ULg

Le SEGI développe dans le campus du Sart Tilman et sur les sites extérieurs de l'ULg une infrastructure d'interconnexion de réseaux locaux départementaux, dont l'objectif est de constituer "l'interréseau de l'ULg" permettant une communication très performante à l'intérieur de l'Université mais aussi vers le monde extérieur.

Le lecteur intéressé par les aspects "réseau" consultera utilement le Guide d'Introduction à l'Interréseau ULg, SEGI, zz01-0117.

## 8.0 Procédures d'exploitation

### 8.1 Tableau des limitations par classe (travaux batch)

Les périodes (A,B,C) caractérisent les différentes exploitations pendant lesquelles les ordinateurs du SEGI sont accessibles aux utilisateurs. Il s'agit:

**A** en permanence (**sauf maintenance**);

**B** en exploitation de type "bâtiment fermé", c'est-à-dire chaque nuit de 22 à 6 heures ainsi que les samedis, dimanches et jours fériés;

**C** selon planification avec le bureau du contrôle des opérations.

Pour raison de maintenance, il est prévu:

1. un arrêt systématique des environnements VM chaque premier vendredi du mois de 18 à 22h;
2. une possibilité d'arrêt partiel des environnements VM chaque vendredi de 18 à 20h lorsqu'une intervention "urgente" ne peut attendre le premier vendredi du mois;
3. une possibilité d'arrêts plus conséquents certains week-end pour permettre des entretiens exceptionnels (installations de matériel ou modifications système plus importantes).

Tous les arrêts d'exploitation sont préalablement annoncés aux utilisateurs par message de BROADCAST qui peut être affiché en cours de session par l'utilisateur de la commande CP Q LOGMSG (à mettre utilement dans le PROFILE EXEC exécuté au LOGON).

De plus, durant chaque période de tests, la dernière ligne des écrans (3270 ou émulation) affiche l'information TEST... en lieu et place de BLIULGxx. Les utilisateurs sont ainsi prévenus qu'ils travaillent dans un environnement "dangereux" et qu'il y a lieu de clôturer leur session. Rappelons en effet que, durant ces périodes de "maintenance", les sessions peuvent être **interrompues à tout instant** sans mise en garde particulière de l'opérateur, avec les conséquences qui peuvent en résulter comme la perte d'un fichier, d'un job ou d'un minidisque.

#### REMARQUES IMPORTANTES

1) Les périodes d'utilisation des classes de travaux batch sont distinctes des périodes de facturation. Pour rappel, le SEGI pratique un tarif de nuit pour les ressources "CPU" et "SIO" consommées :

- entre 18 h et le lendemain matin 7 h, les jours ouvrables;
- les samedis, dimanches et autres jours de fermeture.

C l a s s e	Nb max de procs en //	CPU max	R é g i o n	Nb lignes max	Bandes et cassettes		V e c t e u r	P é r i o d e	Mdisk de travail A
					Jobs max	Setup max /job			
Z	1	20"	4MB	10000	0	0	N	A	1 cyl (195)
A	1	30"	4MB	750000	1	1	Y	A	20 cyl (195)
D	1	10'	9MB	750000	1	2	Y	A	40 cyl (195)
E	2	10'	100MB	750000	1	2	Y	A	40 cyl (195)
N	1	ill.	14MB	750000	0	0	Y	B	80 cyl (195)
G	2	ill	11MB	750000	1	3	Y	C	80 cyl (195) 2654 cyl (200)
H	2	ill	1000MB	750000	1	3	Y	C	80 cyl (195) 2654 cyl (200)

2) En fonction de la charge des équipements, les travaux de classe N peuvent être autorisés plus tôt dans la soirée, mais toujours après 18 heures. Selon les ressources qu'il immobilise, un job de classe N toujours actif à 7 h du matin, pourra voir son exécution suspendue; dans ce cas, il redémarrera à la prochaine période d'exploitation de type "bâtiment fermé".

3) Les travaux de classe G et H sont réservés à des utilisations exceptionnelles. De ce fait, ils ne seront planifiés qu'après accord du "contrôle des opérations". Pour l'accès au minidisque 200 de 2654 cyl., on utilisera le paramètre TDISK MAX de la commande SENBATCH (cf. HELP SEGI SENBATCH)

4) Les travaux de classe N, G et H sont limités à l'exécution de programmes importants sous forme de TEXT's ou MODULE's; il n'est donc pas possible d'utiliser ces classes, par exemple, pour des compilations. D'autre part, vu l'absence de limite CPU pour ces classes, il est conseillé à leurs utilisateurs d'employer la carte /SET TIME ou le paramètre SECONDS de la commande SENBATCH afin de se prémunir contre un éventuel bouclage. Par ailleurs, l'absence de limite CPU classe les jobs concernés dans la catégorie des travaux "CPU bound" dont le démarrage peut-être retardé si la charge machine le justifie.

5) Les travaux de classe E et H sont soumis respectivement aux mêmes règles d'exploitation que ceux des classes D et G. Ces classes sont réservées aux exploitations en mode XA qui utilisent l'adressage étendu et demandent des tailles mémoire supérieures à 16 Mbytes.

## 8.2 Utilisation des périphériques réels

1. Imprimante rapide LASER SIEMENS, type 2200-3 (compatible IBM 3800-3)

L'imprimante SIEMENS, d'une vitesse de 100 pages par minute, offre une résolution de 240x240 "dots per square inch" et est utilisée actuellement en mode "ligne". Elle est réservée aux éditions effectuées sur papier standard (8.5"x322 mm) et accepte tous les fichiers créés dans les classes A et T avec un FORM 0001 (valeur par défaut); une page permet d'imprimer 68 lignes de 132 positions chacune.

2. Imprimante à chaîne de marque MEMOREX, type 4304 (compatible IBM 3203-5)

Elle accepte les impressions "spéciales" qui font appel au montage de documents spéciaux; il s'agit de fichiers produits en classe A ou T pour lesquels l'utilisateur code le paramètre FORM dans la commande CP SPOOL. Les principaux papiers spéciaux sont:

**FORM 0002** recyclé, 8"x375 mm, 8 l/pouce

**FORM 0110** étiquettes, 3 de front

Remarques:

Le FORM 0001 n'est utilisable que sur la LASER SIEMENS.

Les documents spéciaux qui ne pourront être imprimés avant 24 heures seront traités, en priorité, le lendemain matin.

Afin d'éviter un engorgement des files d'attente, les fichiers créés par mégarde dans une classe non autorisée seront détruits par une action de l'opérateur.

3. Imprimante LASER type IBM 3812 (voir «Impression en format A4: imprimante 3812» à la page 60).

4. Autres unités :

- a. les cassettes et les bandes magnétiques sont utilisables par les programmes batch uniquement. Le tableau des limitations donne le nombre maximum de dérouleurs autorisés par classe de travail.
- b. la table traçante CALCOMP étant connectée "online" permet d'effectuer les tracés les uns après les autres, au fur et à mesure de leur création. La découpe et la restitution des tracés est toutefois soumise aux règles suivantes:

- les tracés normaux (avec pointes "floating ball") sont découpés avant chaque départ de la tournée camionnette;
- les tracés à la plume sont effectués chaque jour en fin de "flow" et découpés au plus tard pendant la pause du matin de façon à pouvoir être restitués à la tournée camionnette de 12 heures.

## 8.3 Accès via le réseau commuté

Le réseau commuté est accessible en composant le numéro 041/66.45.45<sup>22</sup>. Ce numéro donne accès à plusieurs lignes du réseau PBX sur lesquelles sont acceptées les vitesses suivantes :

- 300 bps - norme CCITT V21
- 1200 bps - norme CCITT V22
- 2400 bps - norme CCITT V22 bis
- 9600 bps - norme CCITT V32

Les accès aux ordinateurs du SEGI en émulation 3270 ou en mode "ligne à ligne" ainsi que les accès à l'intraréseau s'effectueront en demandant les "services" PBX appropriés (cfr. «Concentrateur PBX» à la page 82).

## 8.4 Procédures de back-up système

Un sauvetage hebdomadaire des volumes contenant les minidisques des utilisateurs est effectué par le SEGI. Précisons qu'il s'agit d'une opération de "dump" physique non adaptée à la récupération de fichiers pris individuellement ou de minidisques, mais bien à celle de disques complets. Ces copies de sécurité sont donc inaccessibles à l'utilisateur et utilisées par le SEGI en cas de destruction totale d'un disque magnétique.

Cette procédure ne remplace nullement la sécurité que chaque utilisateur peut obtenir en utilisant le **SPACE MANAGER** pour la sauvegarde de ses propres fichiers.

## 8.5 Conservation des fichiers dans un spool VM

Pour éviter une saturation de l'espace disque réservé au SPOOL et éviter son engorgement par des fichiers "oubliés" par les utilisateurs, un contrôle journalier de cet espace conduit aux actions suivantes:

1. Scratch des fichiers de type "punch" créés depuis plus de 7 jours;

<sup>22</sup> Des réorganisations du réseau de télétraitement peuvent imposer une redistribution des numéros d'appel. De ce fait, les utilisateurs intéressés seront attentifs aux nouvelles affectations communiquées par la voie de SEGI-INFORMATIONS.

2. Scratch des fichiers de type "print" créés depuis plus de 7 jours;

3. Scratch des fichiers de type "reader" créés depuis plus de 6 jours<sup>23</sup>.

---

<sup>23</sup> Les fichiers de type "reader" en provenance du réseau "EARN" sont conservés 40 jours.



## 9.0 Logiciels d'applications

Les principaux logiciels d'applications actuellement disponibles au SEGI sont les suivants:

- le **S.A.S.** (Statistical Analysis System): logiciel pour l'analyse statistique et économétrique, la recherche opérationnelle, la gestion de données, le calcul matriciel, la programmation d'applications graphiques;
- la **NAG FORTRAN LIBRARY** du "Numerical Algorithm Group";
- le **BMDP** (Biomedical Computer Programs);
- le **GLIM** (Generalised Linear Interactive Modelling);
- le **TSP** (Time Series Processor);
- le logiciel interactif de statistique **STATPACK**;
- les logiciels graphiques **CA-DISSPLA** (logiciel général pour les problèmes graphiques), **GDDM** (Graphical Data Display Manager) et **CALCOMP** (bibliothèque de sous-programmes pour l'utilisation de la table traçante CALCOMP);
- le **SQL** (Structured Query Language), logiciel de gestion de base de données de type relationnel;
- le **Search Manager**, logiciel de recherche documentaire;
- le **DCF** (Document Composition Facility), logiciel de traitement de textes;
- la **PUBLIB** (Public Library) qui contient un ensemble de programmes et sous-programmes de nature essentiellement utilitaire;
- le **ISPF/PDF**, package qui permet d'exploiter les caractéristiques techniques des terminaux IBM 3270 et compatibles;
- le **REDUCE**, logiciel de calcul symbolique.

Le tableau suivant reprend les logiciels cités ci-dessus et indique à laquelle des trois catégories suivantes ils appartiennent :

**"370 only"** (ou "compatibility mode") : le programme ne peut s'exécuter que dans une machine virtuelle d'architecture 370; c'est le cas des programmes qui utilisent des instructions propres à l'architecture 370.

**"XA toleration"** : le programme peut s'exécuter dans une machine virtuelle d'architecture 370 aussi bien que XA mais il ne profite pas pleinement des capacités particulières de l'architecture XA, parce qu'il manipule des adresses de 24 bits.

**"XA exploitation"** : le programme s'exécute dans une machine virtuelle d'architecture XA et est capable de profiter de toutes les capacités de cette architecture, en particulier d'un espace adressable qui s'étend jusqu'à 2Gbytes; il manipule pour ce faire des adresses de 31 bits.

Logiciel	Classification XA
SAS	XA toleration
NAG	XA exploitation
BMDP	XA toleration
GLIM	XA toleration
TSP	XA toleration
STATPACK	XA toleration
DISSPLA	XA toleration
GDDM	XA exploitation
CALCOMP	XA toleration
SQL	XA exploitation
Search Manager	XA toleration
DCF	XA exploitation
PUBLIB	XA toleration
ISPF/PDF	XA exploitation
REDUCE	XA toleration

### 9.1 SAS

#### 9.1.1 Introduction.

**SAS**, "Statistical Analysis System" est un logiciel d'analyse statistique, économétrique et de recherche opérationnelle qui possède de puissants outils pour la gestion des données, le calcul matriciel et la programmation d'applications graphiques. Ses procédures de gestion de données et de calcul matriciel font que SAS est aussi un véritable langage de programmation.

Au SEGI, SAS comprend les produits suivants:

- SAS/BASE** Gestion de données (transformations de données, création de nouvelles variables, tris, sélections, découpe en sous-fichiers, fusion de plusieurs fichiers, ...).
- SAS/STAT** procédure d'analyse statistique.
- SAS/ETS** Procédures d'analyse économétrique.
- SAS/OR** Procédures de recherche opérationnelle.
- SAS/IML** Langage matriciel interactif
- SAS/GRAPH** Procédures graphiques pour tables traçantes et terminaux graphiques.
- SAS/FSP** Procédures de gestion de terminaux de type 3270 et compatibles (utilisation du mode "fullscreen"; acquisition, consultation, mise à jour de données par menus; consultation de fichiers de résultats; ...).
- SAS/CALC** Tableur 3D.
- SAS/CONNECT** Traitement coopératif entre sessions SAS.
- SAS/ASSIST** Utilisation du SAS en mode assistant.

### 9.1.2 Fichiers SAS.

SAS manipule des fichiers qui ont un format particulier. Un fichier SAS se présente sous forme d'une table dans laquelle les colonnes correspondent à des variables et les lignes à des observations.

Exemple: Le fichier étudiants dans lequel on a pour chaque enregistrement les variables 'SECTION', 'NOM' et 'COTE':

SECTION	NOM	COTE
A	CARTER	15
A	DUPONT	12
B	DURAND	16
B	HAMILTON	11
B	VANWEE	13

Pour nommer un fichier en SAS, on utilise un nom à deux particules séparées par un point: la première particule est le nom de librairie (lname), elle correspond au filetype d'un fichier CMS; la seconde particule est un nom de membre (mname), elle correspond au filename d'un fichier CMS. Exemple, le fichier étudiants pourrait être nommé comme suit:

```
nom du fichier SAS: SASFILE.ETUD
nom du fichier CMS correspondant: ETUD SASFILE A
```

### 9.1.3 Programme SAS.

Il y a trois façons de travailler en SAS:

1. En mode assistant (taper la commande "SASASSIS"): c'est la façon la plus simple de travailler. Elle est recommandée pour les débutants. L'utilisateur est guidé dans

son travail par une série de menus; il lui suffit généralement de répondre à des questions ou de sélectionner des options.

2. La façon conversationnelle ou interactive: on entre les instructions SAS au terminal à mesure que SAS les demande; ces instructions sont exécutées immédiatement.

En interactif, il est possible de travailler soit "en mode ligne" (taper la commande "SAS (NODMS)") soit sous contrôle d'un gestionnaire d'écran appelé DMS (taper la commande "SAS"). Dans ce dernier cas, il faut obligatoirement un terminal 3270 ou compatible (voir documentation).

3. La façon différée: on constitue d'abord un fichier qui contient les instructions du programme SAS avant de le faire exécuter. C'est obligatoirement la façon de procéder quand on travaille en CMS BATCH. C'est aussi cette façon de procéder qui est décrite dans les deux exemples qui suivent.

#### 9.1.3.1 Exemple 1.

Créer un fichier SAS à partir des données du fichier étudiants.

**Première solution.** Utiliser l'éditeur XEDIT pour écrire le programme SAS dans le fichier PROG1 SAS par exemple:

```
/* PROGRAMME PROG1 SAS */
DATA SASFILE.ETUD;          /* 1*/
  INPUT SECTION $ NOM $ COTE; /* 2*/
  CARDS;                     /* 3*/
  A      CARTER      15
  A      DUPONT      12
  B      DURAND      16
  B      HAMILTON    11
  B      VANWEE      13
```

Commentaires:

- Les /\* et \*/ encadrent des commentaires dans le programme.
- L'instruction DATA (/\* 1\*/) indique le nom du fichier SAS qui doit être créé. Elle marque aussi le début d'un paragraphe. Un programme SAS peut comprendre un ou plusieurs paragraphes dont le début est indiqué par une instruction DATA ou une instruction PROC. Un paragraphe se termine dès qu'un autre commence. Le programme précédent comprend un seul paragraphe.
- L'instruction INPUT permet de nommer les variables du fichier. Les variables peuvent être de type numérique ou alpha-numérique; dans ce dernier cas, le nom de la variable doit être suivi du signe \$. Dans le cas présent, les données sont lues en format libre (ce qui correspond au "list-directed" de PL/1); il suffit que deux données consécutives soient séparées par un blanc au moins. L'instruction INPUT pourrait aussi contenir des spécifications de format.

- L'instruction CARDS (*/\* 3\*/*) annonce que les données à introduire dans le fichier SAS suivent. Les données se terminent à la fin du fichier ou à la ligne qui précède le premier point-virgule rencontré après l'instruction CARDS.

Pour déclencher l'exécution de ce programme, taper le commande

```
[EXEC] SAS PROG1 (LT
```

**Figure 165. Commande EXEC SAS PROG1**

L'option LT (LType) permet de faire apparaître les messages concernant l'exécution du programme au terminal (ce que l'on appelle le fichier SASLOG). Pour une description complète de la procédure de commandes SAS, voir HELP SEGI SAS.

**Remarque:** dans l'exemple précédent, les données à introduire dans le fichier SAS sont définies dans le programme SAS lui-même. Ces données pourraient avoir été préalablement enregistrées sur disque dans un fichier CMS, ETUD DATA A par exemple. Dans ce cas, le programme de création du fichier SAS pourrait être le suivant:

```
/* PROGRAMME PROG2 SAS */
DATA SASFILE.ETUD;
INFILE IN 'ETUD DATA A';      /* 1*/
INPUT SECTION $ NOM $ COTE;    /* 2*/
```

Commentaires:

L'instruction */\* 1\*/* définit le fichier d'entrée du programme SAS.

L'instruction INPUT (*/\* 2\*/*) lit les données dans le fichier d'entrée, en format libre comme dans l'exemple précédent.

**Deuxième solution.** Acquérir les données par menus sur un terminal 3270 ou compatible. On commence par définir le fichier SAS et les variables qu'il contiendra avec leur type et leur longueur au moyen d'un programme qui pourrait être:

```
/* PROGRAMME PROG3 SAS */
DATA SASFILE.ETUD;
LENGTH SECTION $ 1 NOM $ 15 COTE 2; /* 1*/
STOP;                                  /* 2*/
```

Commentaires:

L'instruction LENGTH (*/\* 1\*/*) définit les variables, leur type (présence ou absence du caractère \$) et leur longueur (nombre de caractères maximum pour les variables alpha-numériques et nombre de bytes pour les variables numériques).

L'instruction STOP (*/\* 2\*/*) est obligatoire; ici, on crée un fichier vide.

Après avoir exécuté le programme précédent, on pourra faire appel à l'éditeur "fullscreen" du SAS au moyen du programme SAS suivant:

```
/* PROGRAMME PROG4 SAS */
PROC FSEDIT DATA=SASFILE.ETUD;
RUN;
```

En exécutant ce programme, on recevra à l'écran un menu sur lequel on retrouvera les noms des variables du fichier SASFILE.ETUD et qu'il suffira de compléter pour entrer les données.

L'éditeur SAS utilise principalement les touches fonctions suivantes:

```
PF3: quitter l'éditeur et sauver le fichier.
PF7: revoir l'observation précédente.
PF8: voir l'observation suivante.
PF9: ajouter une nouvelle observation.
PF10: revoir l'écran précédent de la même observation.
Utile quand une observation exige plus d'un écran.
PF11: voir l'écran suivant de la même observation.
```

### 9.1.3.2 Exemple 2.

Le fichier des données SASFILE.ETUD étant créé, calculer les moyennes et déviations standards des cotes d'étudiants dans chaque section. Soit le programme SAS:

```
/* PROGRAMME PROG5 SAS */
PROC SORT DATA=SASFILE.ETUD; /* 1*/
BY SECTION;                  /* 2*/
PROC MEANS;                   /* 3*/
BY SECTION;                   /* 4*/
```

Commentaires.

Le programme précédent comprend deux paragraphes dont le début est identifié par les instructions PROC (*/\* 1\*/* et */\* 3\*/*).

Le premier paragraphe fait appel à la procédure SAS de tri. Le fichier SASFILE.ETUD est trié dans l'ordre des sections (instruction */\* 2\*/*). Ce tri est nécessaire pour le calcul, par sections, des moyennes et déviations standards.

Le deuxième paragraphe fait appel à la procédure SAS MEANS qui permet le calcul de toute une série d'indicateurs statistiques dont la moyenne et la déviation standard. Ces indicateurs seront calculés pour chaque section (instruction */\* 4\*/*).

Pour exécuter le programme précédent, taper la commande

```
[EXEC] SAS PROG5 (LT PD)
```

**Figure 166. Commande EXEC SAS PROG5**

La procédure SAS engendre généralement deux fichiers CMS qui, dans le cas présent, s'appelleront PROG5 SASLOG, fichier des messages relatifs à l'exécution du programme, et PROG5 LISTING, fichier des résultats produits par les procédures SAS appelées. Les options suivantes définissent la destination de chacun de ces fichiers:

```
SASLOG LISTING DESTINATION
-----
LType PType Console virtuelle
LPrint PPrint Imprimante virtuelle
LDisk PDisk Disque
```

Dans le cas présent, le fichier SASLOG est envoyé à la console et le fichier LISTING, sur disque. Ce dernier fichier pourra être affiché, édité ou imprimé au moyen des commandes CMS TYPE, XEDIT ou PRINT respectivement.

### 9.1.4 Remarques.

1. Pour une description complète de la procédure SAS, voir HELP SEGI SAS.
2. SAS comprend un grand nombre d'interfaces permettant de piloter des matériels graphiques - voir la brochure *SAS/GRAPH au SEGI*, SEGI, zz01-0109.
3. Il existe un interface entre SAS et BMDP. Voir HELP SEGI SASBMDP et la procédure BMDP en SAS.
4. Le SEGI et le Service de Mathématiques Appliquées de l'Institut de Psychologie et des Sciences de l'Education, ULG, ont développé une procédure SAS de calcul d'indices de proximités ou dissimilarités sur des matrices de mesures, d'occurrences, disjonctives, de rangs ou mixtes. Voir "La procédure SAS DISSIM", SEGI, ZZ01-0101.
5. Les noms suivants sont réservés comme noms de librairie par SAS et ne peuvent donc être employés librement par l'utilisateur:

```
SAS        WORK
SASLIB     LIBRARY
SASUSER    USER
SASHELP    FTnnF001
```

Les noms de bibliothèques commençant par TAPE sont réservés pour des fichiers en "format bande magnétique" créés avec la version 5 de SAS.

6. SAS accède sans problème aux fichiers SAS qui se trouvent sur le mini-disque A. Pour accéder à des fichier SAS se trouvant sur un mini-disque autre que A, il faut utiliser l'instruction SAS suivante :

```
LIBNAME lname 'fm'
```

"lname" désigne le nom de librairie du fichier SAS; fm est le filemode du mini-disque concerné.

7. SAS choisit, pour ses fichiers temporaires de travail (PARMCARDS, WORK, format LIBRARY), le filemode du mini-disque accessible en lecture-écriture qui possède le plus grand espace libre. On peut imposer ce filemode grâce à l'option SIODISK fm de la procédure SAS.
8. *Attention*, SAS génère une commande "ERASE \* WORK fm" où fm désigne le filemode choisi pour les fichiers de travail SAS.
9. Le SAS utilise le processeur vectoriel si l'option VECTOR est activée. Au SEGI, le processeur vectoriel n'est accessible qu'en batch.

## 9.2 La NAG FORTRAN Library.

La NAG FORTRAN LIBRARY est un ensemble de sous-programmes FORTRAN qui portent sur les sujets suivants:

```
COMPLEX ARITHMETIC
ZEROS OF POLYNOMIALS
ROOTS OF ONE OR MORE TRANSCENDENTAL EQUATIONS
SUMMATION OF SERIES
QUADRATURE
ORDINARY DIFFERENTIAL EQUATIONS
PARTIAL DIFFERENTIAL EQUATIONS
NUMERICAL DIFFERENTIATION
INTEGRAL EQUATIONS
INTERPOLATION
CURVE AND SURFACE FITTING
MINIMIZING AND MAXIMIZING A FUNCTION
MATRIX OPERATIONS, INCLUDING INVERSION
EIGENVALUES AND EIGENVECTORS
DETERMINANTS
SIMULTANEOUS LINEAR EQUATIONS
ORTHOGONALISATION
SIMPLE CALCULATIONS ON STATISTICAL DATA
CORRELATION AND REGRESSION ANALYSIS
ANALYSIS OF VARIANCE
RANDOM NUMBER GENERATORS
UNIVARIATE ESTIMATION
NONPARAMETRIC STATISTICS
CONTINGENCY TABLE ANALYSIS
TIME SERIES ANALYSIS
OPERATIONS RESEARCH
SORTING
ERROR TRAPPING
APPROXIMATIONS OF SPECIAL FUNCTIONS
MATHEMATICAL CONSTANTS
MACHINE CONSTANTS
```

## INNERPRODUCTS INPUT/OUTPUT UTILITIES

Une documentation enregistrée donne la liste des sous-programmes, leurs fonctions respectives et leur notice d'utilisation. La documentation complète est disponible au bureau de documentation du SEGI (NAG FORTRAN LIBRARY Manuals). Seuls sont disponibles les codes objets des différents programmes.

Deux versions des bibliothèques NAG sont disponibles :

- une version **scalaire** : NAGSLIB1 et NAGSLIB2 accessibles via la procédure de commandes NAG;
- une version **vectorisée** : NAGVLIB1 et NAGVLIB2 accessibles via la procédure de commandes NAGV. Avec la version vectorisée, les bibliothèques ESSL d'IBM (ESVVLIBA TXTLIB et ESVVLIBB TXTLIB) sont aussi nécessaires et, dans la commande GLOBAL TXTLIB, doivent être spécifiées avant les bibliothèques NAG. Pour ESSL, voir «VS FORTRAN» à la page 49.

Procédure d'accès au disque de la machine PA qui contient les bibliothèques:

```
[EXEC] NAG  
ou  
[EXEC] NAGV
```

**Figure 167. Commande EXEC NAG**

La commande NAG engendre les commandes CMS suivantes qui rendent accessibles les bibliothèques scalaires de ESSL et NAG :

```
GLOBAL TXTLIB VSF2LINK VSF2FORT ESVSLIBA ESVSLIBB  
NAGSLIB1 NAGSLIB2 CMSLIB  
GLOBAL LOADLIB
```

La commande NAGV engendre les commandes CMS suivantes qui rendent accessibles les bibliothèques vectorisées de ESSL et NAG :

```
GLOBAL TXTLIB VSF2LINK VSF2FORT ESVVLIBA ESVVLIBB  
NAGVLIB1 NAGVLIB2 CMSLIB  
GLOBAL LOADLIB
```

Après avoir exécuté l'une de ces commandes, l'utilisateur peut exécuter un programme qui fait appel à un ou plusieurs sous-programmes des bibliothèques mentionnées.

Procédure de consultation de la documentation NAG enregistrée:

```
[EXEC] NAGHELP
```

**Figure 168. Commande EXEC NAGHELP**

La commande

```
HELP SEGI NAGHELP
```

donne accès au manuel d'utilisation de la commande NAGHELP.

### 9.3 *Le système BMDP.*

Le système BMDP (Biomedical Computer Programs) comprend l'ensemble des programmes BMDP, une documentation relative à ces programmes, les procédures d'utilisation et des jeux de données de test.

La documentation comprend:

- Un catalogue enregistré des programmes (HELP BMDP);
- Un manuel d'utilisation disponible au bureau de documentation du SEGI <sup>24</sup>.

Seul le code objet de chaque programme est disponible.

Procédure d'exploitation:

```
[EXEC] BMDP [fileid] (PROG progid [ DISK  
PRINT ] [AUTO  
TERM ]  
[TRANSF fname1] [FUNCTION fname2] [SAVEIN fname3]  
[SAVEOUT fname4]
```

**Figure 169. Commande EXEC BMDP**

- **fileid** permet d'identifier un fichier qui contient les commandes BMDP. Ce fichier doit être utilisé si le programme est exécuté en différé; si ce fichier n'est pas spécifié, le programme est exécuté de façon interactive.

fileid a la forme suivante:

<sup>24</sup> BMDP Statistical Software, University of California (2 volumes).

```
filenameBMDP *[filetype filemode]
```

- **progid** est le nom du programme tel qu'il apparaît dans le catalogue ou abrégé en ses 2 ou 3 derniers caractères (exemple: BMDPID ou PID ou ID).
- Les options **DISK**, **PRINT (ou PRT)**, **TERM (ou TYPE)** permettent d'indiquer la destination de l' "output" du programme:

Si **DISK** est spécifié, l' "output" est placé dans un fichier dont le nom est:

- " @@@ LISTING A1 " si le programme est exécuté de façon interactive;
- filename LISTING A1 si l'option fileid avait été spécifiée; **DISK** est l'option par défaut dans ce second cas.

Si **PRINT** est spécifié, l' "output" est envoyé vers l'imprimante de la machine virtuelle de l'utilisateur.

Si **TERM** est spécifié, l' "output" est envoyé à la console de l'utilisateur; **TERM** est l'option par défaut quand le programme est exécuté de manière interactive.

- **fname1 et fname2** désignent des fichiers dont le filetype doit être FORTRAN, qui peuvent se trouver sur n'importe quel disque CMS et qui permettent de définir, respectivement, des transformations FORTRAN ou des fonctions utilisées par le programme. Comme les options TRANSF et FUNCTION font appel au compilateur FORTRAN, leur emploi n'est autorisé qu'en CMS batch.
- **AUTO** est une option du "CMS loader" qui peut être utile lorsque les options TRANSF et/ou FUNCTION sont spécifiées. **AUTO** permet au "LOADER" de balayer les disques de la machine de l'utilisateur pour résoudre les références externes au moment du chargement du programme. Dans la procédure BMDP, l'option par défaut est NOAUTO.
- **fname3 et fname4** désignent des fichiers dont le filetype doit être BMDPSAVE.

fname3 désigne un "input save file" qui doit se trouver sur le disque A et qui sera automatiquement associé à l'unité FORTRAN 3;

fname4 désigne un "output save file" qui existe ou qui sera créé sur le disque A et qui est associé à l'unité FORTRAN 4.

Jeux de données de test:

Tout programme BMDP peut être exécuté comme suit sur le jeu de données de test correspondant:

```
[EXEC] BMDP progid (PROG progid...
```

## 9.4 Le logiciel GLIM

Le GLIM ("Generalised Linear Interactive Modelling") est un programme interactif d'analyse statistique.

Il peut être appelé via la commande

```
[EXEC] GLIM [fn [(options)]]
```

Figure 170. Commande EXEC GLIM

La commande et les options sont décrites dans HELP SEGI GLIM.

Documentation:

The Generalised Linear Interactive Modelling System, Manual.

## 9.5 Le logiciel TSP

Le TSP ("Time Series Processor") est un logiciel d'analyse économétrique utile dans les domaines suivants :

- prévisions de ventes;
- recherches et prévisions dans le domaine macroéconomique;
- analyse financière;
- analyse et prévision de coûts;
- analyses économétriques diverses.

Il peut être appelé via la commande

```
[EXEC] TSP [fn [fm]]
```

Figure 171. Commande EXEC TSP

fn est le "filename" d'un fichier dont le "filetype" est TSP et qui contient le programme qui doit être exécuté. fm est le "filemode" du disque sur lequel sera envoyé le fichier "fn LISTING" qui contient les résultats. Par défaut, "fm" sera le "filemode" du minidisque accessible en écriture qui a le plus grand espace libre.

Une seconde commande est disponible :

```
[EXEC] XTSP
```

Figure 172. Commande EXEC TSP

elle permet d'appeler le logiciel TSP en mode "FILELIST" pour éditer et exécuter un programme TSP qui existe sur le minidisque A.

Après avoir tapé la commande FILELIST, tapez XTSP dans la colonne "Cmd" en regard du fichier de type TSP qui contient votre programme. Votre fichier est édité; vous pouvez le modifier. Dès que vous quittez l'éditeur, l'exécution du programme démarre.

Les commandes TSP et XTSP sont décrites respectivement dans

```
HELP SEGI TSP
HELP SEGI XTSP
```

Documentation :

Time Series Processor, Reference Manual.

Time Series Processor, User's Manual.

## 9.6 *Le logiciel STATPACK*

STATPACK est un logiciel interactif de statistique développé par le Professeur J. K. LINDSEY. Les principaux domaines couverts sont les suivants:

- Statistique descriptive.
- Histogrammes.
- Diagrammes cartésiens.
- Corrélation.
- Analyse de variance.
- Régression linéaire.
- Analyse de covariance.
- Tables de fréquences.
- Ajustements logistiques et log-linéaires.
- Ajustements de distributions discrètes ou continues à des données réparties en classes.
- Diverses fonctions de manipulations et de transformations de données.

Une liste complètes des fonctions disponibles peut être obtenues par HELP SEGI STATPACK.

Pour faire appel au logiciel, il suffit de taper la commande STATPACK.

## 9.7 *Applications graphiques.*

Les différents logiciels graphiques disponibles au SEGI peuvent être classés en deux catégories:

1. logiciels destinés à une unité périphérique d'un type bien déterminé:
  - a. GDDM et GAM pour le matériel IBM,
  - b. CALCOMP pour la table traçante.
2. logiciels à large spectre de périphériques (CA-DISSPLA et SAS) permettant à l'utilisateur de construire des ordres graphiques formellement indépendants de l'unité utilisée.

Les possibilités de SAS <sup>25</sup> s'inscrivent dans un cadre plus large décrit ailleurs aussi nous ne reprendrons ci-dessous que les consignes d'utilisation des logiciels spécifiquement graphiques, à savoir:

1. la bibliothèque des sous-programmes CALCOMP accessible via les procédures CALCOMP pour les dessins banalisés avec pointes traçantes standard, ou CALCOMPX pour les dessins spéciaux à l'encre de Chine.
2. le système CA-DISSPLA accessible via la procédure DISSPLA.
3. GDDM

**Remarque :** Les procédures citées donnent accès au minidisque de la machine PA qui contient les fichiers des systèmes graphiques et peuvent modifier certaines options de la console virtuelle de l'utilisateur (cfr commandes SET et TERMINAL).

### 9.7.1 CALCOMP.

Il s'agit d'un ensemble de sous-programmes Fortran et Assembler disponibles dans les bibliothèques (TXTLIB) CALCLIB (pour les dessins banalisés avec pointes traçantes standard), ou CALCLIBX (pour les dessins spéciaux à l'encre de Chine).

De plus, une bibliothèque d'interface (TXTLIB) BENCAL permet de continuer à exploiter sur la table CALCOMP des programmes qui utilisaient le logiciel de l'ex-table Benson.

Les procédures CALCOMP/CALCOMPX sauvent, par la procédure GLPUSH, la liste des TXTLIB et LOADLIB en vigueur, et génèrent les commandes CMS suivantes:

```
GLOBAL TXTLIB CALCLIB/CALCLIBX BENCAL VSF2LINK VSF2FORT
GLOBAL LOADLIB
```

Le chargement et l'exécution du programme de l'utilisateur sont assurés par la procédure CALGO nom-du-programme nn où nn désigne le dname Fortran (FTnnF001) du fichier de sortie Calcomp (par défaut, nn=02).

---

<sup>25</sup> SAS/GRAPH User's guide.

Cette procédure perfore le fichier graphique de sortie dans le lecteur d'une machine dénommée CALCOMP (ce qui interdit l'utilisation simultanée du punch à d'autres fins) qui planifie le tracé effectif sur la table traçante.

L'utilisateur qui ne désirerait pas utiliser cette procédure peut s'en inspirer librement; impérativement toutefois, le fichier graphique doit parvenir à la machine CALCOMP en classe B et ne peut y être doté d'un header.

## 9.7.2 DISSPLA

DISSPLA est un logiciel graphique distribué par Computer Associates.

Il se présente sous forme d'un ensemble de routines, réparties dans un certain nombre de bibliothèques TXTLIB, appelables dans différents langages dont FORTRAN <sup>26</sup>.

DISSPLA permet de piloter une gamme très étendue de *devices* et comprend principalement des routines graphiques 2D et 3D, des routines de tracé de courbes de niveau, des utilitaires mathématiques (ajustement, interpolation, ...), des possibilités de segmentation, de *graphic input*, de *modeling*, un système intégré de *metafiles*, un générateur d'applications, etc.

Les procédures installées par le SEGI permettent :

- d'accéder au logiciel DISSPLA (commande DISSPLA);
- de charger un programme préalablement compilé en VS FORTRAN (commande DISLOAD).

Trois procédures additionnelles ont été installées :

- DISCALC, pour l'exécution d'un programme qui crée un dessin destiné à la table traçante CALCOMP;
- DISPCTEK, pour l'exécution d'un programme qui crée un dessin destiné à être visualisé sur l'écran d'un PC connecté à l'ordinateur central via le convertisseur de protocole IBM 7171;
- DIS5080, pour l'exécution d'un programme qui crée un dessin destiné à être visualisé sur une station graphique IBM de type 5080.

### 9.7.2.1 La procédure DISSPLA

Cette procédure permet d'accéder aux disques qui contiennent le logiciel DISSPLA.

Syntaxe de la commande :

DISSPLA

Figure 173. Commande EXEC DISSPLA

### 9.7.2.2 La procédure DISLOAD

Cette procédure permet de charger un programme préalablement compilé en VS FORTRAN par la commande FORTVS2.

Syntaxe de la commande :

DISLOAD fn1 [fn2 fn3 ...] [(tn1 tn2 ... [])]

Figure 174. Commande EXEC DISLOAD

- fn1 fn2 ... sont les fichiers TEXT qui doivent être chargés.

**Remarque :** Tous les fichiers nécessaires doivent être spécifiés. Il n'y a pas de chargement automatique.

- tn1 tn2 ... sont des bibliothèques TXTLIB additionnelles.

La commande globalise les bibliothèques FORTRAN et DISSPLA. Si des bibliothèques additionnelles doivent être globalisées, elles doivent être indiquées. Des bibliothèques additionnelles sont notamment indispensables dans les cas suivants :

- Pour l'utilisation de terminaux graphiques IBM supportés en GDDM, les bibliothèques additionnelles ADMNLIB et ADMRLIB doivent être indiquées. La globalisation de ces bibliothèques ne sera évidemment possible que si le logiciel GDDM est accessible; pour le rendre accessible, taper la commande **GDDM**.
- Pour l'utilisation de stations graphiques IBM de type 5080, les bibliothèques additionnelles AFMGLIB, AFMNLIB et AFMRLIB du logiciel GRAPHIGS doivent être indiquées. La globalisation de ces bibliothèques ne sera possible que si le logiciel GRAPHIGS est accessible; pour le rendre accessible, taper la commande **PHIGS**.
- Pour l'utilisation de la table traçante CALCOMP, la bibliothèque additionnelle CALCLIB (ou CALCLIBX si l'on demande un tracé avec plumes plutôt qu'avec pointes standards) doit être indiquée <sup>27</sup>. La globalisation de la bibliothèque CALCOMP adéquate ne sera évidemment possible que si le logiciel CALCOMP est accessible; pour le rendre accessible, taper la commande **CALCOMP**.

<sup>26</sup> Les procédures installées par le SEGI font référence au VS FORTRAN version 2.

<sup>27</sup> Pour des détails sur la table traçante CALCOMP, voir *La Table traçante CALCOMP*, SEGI, ZZ01-0099.

Après avoir chargé un programme au moyen de la commande DISLOAD, on peut

- lancer l'exécution du programme par la commande **START** (ou l'une des commandes **DISCALC**, **DIS5080** ou **DISPCTEK** si le graphique est destiné respectivement à la table traçante CALCOMP, à une station graphique IBM de type 5080 ou à l'écran d'un PC connecté via l'IBM 7171);
- ou créer un fichier module : **GENMOD fn**.

### 9.7.2.3 La procédure DISCALC

La procédure DISCALC peut être utilisée pour envoyer un dessin à la table traçante CALCOMP.

Syntaxe de la commande :

```
DISCALC [fn1 [ft1 [fm1]] [(options [])]]
```

Défauts : ft1=MODULE; fm1=\*

**Figure 175. Commande EXEC DISCALC**

La procédure DISCALC peut être utilisée dans les 3 cas suivants :

1. Pour exécuter un programme qui vient d'être chargé en mémoire virtuelle par DISLOAD et envoyer le graphique à la table CALCOMP. Dans ce cas, il ne faut pas indiquer fn1 ft1 fm1. Syntaxe :

```
DISCALC [(Options [])]
```

2. Pour exécuter un programme préalablement sauvé sur disque sous forme de module et envoyer le graphique à la table CALCOMP. Dans ce cas, il faut indiquer fn1 [ft1 [fm1]]. Syntaxe :

```
DISCALC fn1 [ft1 [fm1]] [(Options [])]
```

La valeur par défaut de ft1 est MODULE.

3. Pour renvoyer à la table CALCOMP un fichier fn PICTURE créé préalablement au moyen de l'option KEEP. Dans ce cas, il faut obligatoirement spécifier le "filetype" PICTURE. Syntaxe :

```
DISCALC fn PICTURE [fm]
```

**Options :** elles ne sont utiles que dans les cas 1 et 2 décrits ci-dessus; dans le cas 3, elles sont ignorées.

- **KEEP fn2 :** permet de conserver le fichier graphique produit par le programme dans un fichier "fn2 PICTURE A". Ce fichier pourra ultérieurement être renvoyé à la table CALCOMP sans plus passer par l'exécution d'un programme (voir cas 3 ci-dessus).
- **UNIT n :** permet de désigner le numéro d'unité FORTRAN à laquelle le graphique est destiné. Sa valeur par défaut est 9.

### 9.7.2.4 La procédure DISPCTEK

La procédure DISPCTEK peut être utilisée pour exécuter, un programme qui crée un dessin destiné à être visualisé sur l'écran d'un PC connecté à l'ordinateur central via le convertisseur de protocole IBM 7171 et le logiciel de communication FTPSEGI<sup>28</sup>. Le PC doit être équipé d'un écran graphique CGA ou EGA.

Syntaxe de la commande :

```
DISPCTEK [fn]
```

**Figure 176. Commande EXEC DISPCTEK**

Si fn n'est pas spécifié, DISPCTEK déclenche l'exécution du programme préalablement chargé en mémoire virtuelle au moyen de la commande DISLOAD. Si fn est spécifié, DISPCTEK déclenche l'exécution de fn MODULE.

### 9.7.2.5 La procédure DIS5080

La procédure DIS5080 peut être utilisée pour exécuter, un programme qui crée un dessin destiné à être visualisé sur l'écran d'une station graphique IBM de type 5080.

Syntaxe de la commande :

```
DIS5080 [fn] [( addr )]
```

**Figure 177. Commande EXEC DIS5080**

Si fn n'est pas spécifié, DIS5080 déclenche l'exécution du programme préalablement chargé en mémoire virtuelle au moyen de la commande DISLOAD. Si fn est spécifié, DIS5080 déclenche l'exécution de fn MODULE.

<sup>28</sup> Le logiciel FTPSEGI, développé par le SEGI, est décrit dans la brochure suivante : *Guide pratique d'utilisation du logiciel FTPSEGI*, zz01-0113. FTPSEGI doit être accompagné du logiciel KERMIT et du fichier MSKERMIT.TEK.

**addr** désigne l'adresse réelle de la station graphique. Si cette option est indiquée, la procédure d'attachement de la station graphique est déclenchée. La station graphique doit être attachée avant de pouvoir être utilisée. Cette option doit donc être spécifiée au moins lors du premier appel de DIS5080.

**Remarque :** L'adresse réelle de la station graphique est indiquée à la rubrique *Graphic Device Address* du second panneau de *customization* auquel on accède en poussant les touches Alt-Jump sur la station graphique.

La procédure d'attachement qui est appelée est **\$5080\$ EXEC**. Cette procédure peut aussi être appelée en dehors de DIS5080, par exemple dans le PROFILE EXEC; dans ce cas, il n'est plus nécessaire de spécifier l'option **addr** dans le procédure DIS5080.

### 9.7.3 GDDM et GAM

GDDM ("Graphical Data Display Manager") est un programme-produit IBM dont la principale fonction est de gérer les communications entre des programmes d'applications et des terminaux, particulièrement lorsque ceux-ci sont utilisés comme stations graphiques.

GDDM supporte et gère une large gamme de matériels IBM : terminaux, imprimantes et tables traçantes.

Une introduction et la bibliographie complète relatives à GDDM peuvent être trouvés dans "Graphical Data Display Manager, General Information", IBM, GC33-0319.

GAM ("Graphics Access Method") est un programme-produit IBM pour le support des stations graphiques IBM, notamment de type 5080.

Une introduction et la bibliographie complète relatives à GAM peuvent être trouvées dans "Graphics Access Method, General Information", IBM, GC33-0125.

## 9.8 *Le logiciel de recherche documentaire: Search Manager*

Search Manager est un programme produit IBM permettant la recherche documentaire sur du texte intégral.

D'usage aisé, il permet d'effectuer cette recherche en mode conversationnel à partir de mots significatifs du texte. Cette recherche peut être appréhendée également au travers de la logique booléenne, de la technique du voisinage, ou encore à partir de facteurs de pondération. A priori, Search Manager ne requiert aucune programmation et accepte des documents préparés pour DCF. (Voir «PUBLISHING: un environnement complet basé sur les logiciels DCF pour la partie texte et GDDM pour la partie graphique.»)

Pour tout renseignement, consulter le SEGI. Voir aussi «Bibliographie» à la page 113.

## 9.9 *Les bases de données relationnelles: SQL*

Le logiciel SQL est un système de gestion de bases de données relationnelles. Il est accessible via des applications écrites en REXX, FORTRAN, PLI ou COBOL.

Les fichiers SQL sont sous un format particulier et ne peuvent donc être accédés que via ce logiciel. D'autre part un ensemble de verrous permet d'assurer la confidentialité des informations.

Un langage d'interrogations, de mises à jour et de production de rapports appelé QMF permet également, moyennant les autorisations préalables, d'accéder les bases de données SQL directement sous CMS sans passer par l'écriture d'un programme.

Une description du produit est disponible dans la brochure GH09-8044 (SQL/OS Concepts and Facilities) disponible au bureau de documentation du SEGI.

## 9.10 *PUBLISHING: un environnement complet basé sur les logiciels DCF pour la partie texte et GDDM pour la partie graphique.*

Cet environnement permet de mettre en forme des documents<sup>29</sup> de toute nature, de la simple lettre au document très volumineux. Il est particulièrement adapté aux documents issus d'une composition automatique (par exemple des documents partiellement créés via un programme) ainsi qu'aux documents donnant lieu à la création d'une base de données documentaire (un document DCF est compatible avec STAIRS).

Hormis ces créneaux, il est préférable d'utiliser les logiciels de traitement de texte disponibles sur microordinateur.

La commande PUBLISHI permet d'accéder aux différents logiciels IBM permettant de faire de la publication électronique de documents sur l'ordinateur central. Cette commande est publique et accessible à partir de tous les terminaux et micro-ordinateurs connectés au 3090 (noeud BLIULG11).

Elle affiche un menu offrant les possibilités suivantes :

1. Créer de nouveaux documents DCF, éventuellement à partir d'une bibliothèque de documents modèles, en utilisant l'éditeur XEDIT. DCF permet ainsi d'encoder des

<sup>29</sup> Ce manuel a été édité par DCF.

- documents contenant les sous-divisions classiques : page de garde, préface, résumé, table des matières, liste des figures, liste des tableaux, chapitres, sous-chapitres, différents niveaux de titres, appendices, glossaires, annexes, index. DCF permet en outre de créer des listes (simples, numérotées ou avec définitions de termes), d'insérer des figures (graphiques ou non graphiques), des notes de bas de page, des renvois (aux titres, aux notes, aux figures, aux exemples, aux tableaux, aux listes), des tableaux et des formules scientifiques (encodées au moyen d'une notation descriptive). L'utilisateur peut aussi composer le document sur plusieurs colonnes, définir des zones de recouvrement pour superposer des textes et/ou des graphiques, définir son propre jeu d'instructions pour compléter les possibilités offertes en standard, mélanger différents styles de caractères, paramétrer les documents pour les fusionner avec des informations produites par un programme ou issues d'une base de données, demander une vérification orthographique en plusieurs langues à l'intérieur d'un même document, produire des épreuves prévues pour un tirage en recto/verso ou exporter les documents dans une base de données STAIRS pour effectuer des recherches sur du texte intégral.
2. Appeler DCF pour mettre le document en page en spécifiant un fichier d'options. A partir d'un même document source, on dispose ainsi de la possibilité d'effectuer des mises en pages adaptées aux différents types d'impressions ainsi que d'orienter le résultat de la mise en page vers le STAIRS.
  3. Visualiser à l'écran<sup>30</sup> l'épreuve produite par DCF afin de contrôler sa géométrie ou des détails de présentation avant de l'envoyer sur l'imprimante IBM 3812<sup>31</sup>
  4. Scanner des images de pratiquement toute origine (photos, dessins, graphiques) et de les manipuler sur un écran pleine page. Ces manipulations permettent de modifier les proportions ou la taille d'une image et de constituer une image à partir de différents fragments d'images. Les images ainsi digitalisées peuvent ensuite être électroniquement insérées dans un document DCF.
  5. Représenter graphiquement<sup>30</sup> sous différentes formes (histogrammes, courbes, tartes, tours, barres, cartes polaires, diagrammes de Venn... ) des données entrées au clavier ou produites par un programme. Ces graphiques sont automatiquement produits en mode interactif sous ces différentes formes. Ils sont visualisés à l'écran, et on peut ensuite les compléter d'annotations ou effectuer différentes retouches. Le graphique final est sauvé sur disque (le format du graphique ainsi que le graphique lui-même peuvent être sauvés sur disque respectivement dans des fichiers de type ADMCFORM et ADMGDF : le premier pourra ensuite être appliqué à d'autres jeux de données pour produire de nouveaux graphiques, le second pouvant être récupéré dans les documents DCF).
  6. Créer interactivement<sup>30</sup> de nouveaux symboles au moyen de lignes et courbes construites à partir de vecteurs définis par deux points. Les courbes sont définies par minimum deux vecteurs. Ces symboles peuvent ensuite être utilisés dans la composition des graphiques ADMGDF mentionnés auparavant. Cette technique permet notamment de créer des logos ou autres éléments de dessin constituant une boîte à outils pour la composition de dessins plus complexes.
  7. Créer interactivement<sup>30</sup> de nouveaux symboles au moyen de points situés sur une grille repère. Ces symboles ont la même utilité que celle indiquée au point précédent.
  8. Afficher la liste des documents accessibles à l'utilisateur afin qu'il sélectionne celui qu'il désire modifier.
  9. Afficher la liste des documents modèles pour permettre d'en créer de nouveaux ou de modifier ceux qui existent.
  10. Afficher la liste des fichiers d'options pour permettre d'en créer de nouveaux ou de modifier ceux qui existent.
  11. Imprimer un document sur l'imprimante 3812.
  12. Créer un OVERLAY : un document DCF d'une page est ainsi converti en ce qu'on appelle un OVERLAY c'est-à-dire que, lors des impressions sur 3812, on peut demander que les pages d'un document reçoivent comme fond cet overlay. Cette technique permet de supprimer l'usage des préimprimés.
  13. Définir les minidisques sur lesquels résident les images et graphiques qui doivent être imprimés dans les documents envoyés à l'imprimante 3812.
  14. Mettre au point un document<sup>30</sup> par une procédure récursive qui passe par l'édition du document afin de permettre sa modification et qui, lorsqu'on quitte l'éditeur XEDIT, lance automatiquement la mise en page et la visualisation à l'écran de l'épreuve produite. En fait, cette procédure récursive regroupe les points 2, 3 et 8.
  15. Convertir les graphiques ADMGDF dans le format requis par DCF. Ceci permet de les inclure dans un document et de préciser leur taille définitive.
  16. Convertir des graphiques SAS dans le format requis par DCF pour les inclure dans un document et préciser leur taille définitive.
  17. Visualiser à l'écran des graphiques ADMGDF, zoomer et isoler des parties du dessin.

<sup>30</sup> Disponible uniquement sur les terminaux 3270 graphiques.

<sup>31</sup> Pour rappel, la 3812 est une imprimante feuille à feuille qui utilise une technologie comparable au laser. Le présent guide a été imprimé sur la 3812.

## 9.10.1 Quelques exemples

Lorsque vous créez votre fichier DCF (option 2 de la commande PUBLISH) vous devez y insérer des caractères de contrôle, appelés TAGs, qui vont identifier les différentes composantes du document (par exemple les éléments d'une liste).

### 9.10.1.1 Les paragraphes.

Le TAG utilisé pour définir un nouveau paragraphe est **p**. Comme tout TAG il doit être précédé d'un double point (: ) et suivi d'un point ( . ). Ainsi on coderait ce paragraphe comme suit: :p.Le TAG utilisé pour définir un nouveau...

### 9.10.1.2 Les titres.

Les titres identifient les différentes parties du document. Ainsi un titre peut introduire un nouveau chapitre, un nouveau sujet etc.... DCF offre 7 niveaux de titres représentés par les TAGs **h0** à **h6** qui provoqueront 7 manières différentes de mettre le titre en évidence afin de bien marquer la hiérarchie désirée. Chaque titre doit être codé sur une seule ligne avec son TAG. Voici le détail sur ces TAGs:

- h0** permet de diviser le document en plusieurs parties comprenant chacune plusieurs chapitres. h0 sera donc assez peu utilisé.
- h1** peut être utilisé pour introduire un chapitre.
- h2 a h6** introduisent les divisions de niveau inférieur.

### 9.10.1.3 La liste simple.

La liste simple consiste en l'énumération d'une série d'articles. Exemple:

```
deux oeufs
200gr farine
1dl lait
1 pincée de sel
```

elle peut aussi être plus compacte:

```
deux oeufs
200gr farine
1dl lait
1 pincée de sel
```

Une liste simple débute par le TAG 'sl' et se termine par le TAG 'esl'. Chaque élément de la liste doit être précédé du TAG 'li'. Pour la rendre plus compacte on doit l'introduire par le TAG 'sl compact'.

### 9.10.1.4 La liste numérotée.

La liste numérotée est comparable à la liste simple sauf que chaque article sera précédé d'un numéro de séquence. On l'introduit et on la termine par les TAGs 'ol' et 'eol'.

### 9.10.1.5 Vérifier l'orthographe.

Pour demander à DCF de vérifier l'orthographe du document on devra ajouter l'option SPELLCHK.

Par défaut, DCF suppose que le document est en anglais d'où la nécessité pour les francophones de coder également en tête du document le TAG de niveau 1 'dl fnat'. Ce renseignement permettra aussi à DCF de tronquer les mots en utilisant son algorithme français.

### 9.10.1.6 Table des matières.

On peut également terminer le document par le TAG 'toc' pour demander à DCF de produire une table des matières.

### 9.10.1.7 Numéroté les titres.

Les titres de niveau h1 à h4 peuvent être automatiquement numérotés en ajoutant l'option 'SYSVAR(H YES)'.

### 9.10.1.8 Le 'POWER TYPING'.

Si vous préparez les olympiades de la dactylo ou si vous avez un gros document à coder, XEDIT vous offre une alternative au mode INPUT: c'est le power typing. Ce mode de travail permet de considérer l'écran complet comme une seule ligne ce qui signifie que vous pouvez entrer votre texte sans vous préoccuper de la longueur des lignes ni de la longueur des mots.

En d'autres termes vous pouvez entrer votre texte sans même regarder l'écran: dès que le curseur arrive en fin de ligne il passe automatiquement à la ligne suivante et le contenu de l'écran n'est envoyé vers l'ordinateur que lorsque vous enfoncez la touche 'ENTER'. A ce moment XEDIT met le texte dans le fichier en évitant les tronçatures de mots. Si vous voulez forcer un saut à la ligne il vous suffit de coder le signe # à l'endroit désiré: ce dernier sera interprété par XEDIT comme un saut de ligne et n'apparaîtra donc pas dans le fichier.

On entre en mode POWER TYPING au moyen de la commande 'POWER' (en lieu et place de 'INPUT') et on en sort en faisant deux ENTERS successifs. Ce mode de travail

permet d'utiliser le terminal 3270 à la vitesse de la lumière (ou presque) quelque soit la charge de l'ordinateur ou de la ligne.

### 9.10.1.9 Tableaux

Pour illustrer les possibilités de tableaux, voici la liste des options de la commande PUBLISHI disponibles sur chaque type de terminaux.

Fonctions	3270 typoriche	3270 graphique	3193 & scanner
Création de document	X	X	X
Formatage de document	X	X	X
Visualisation de document	X	X	X
Manipulation d'images			X
Edition de symboles		X	
Edition de cartes		X	
Modification de documents	X	X	X
Edition de modèles	X	X	X
Définition d'options	X	X	X
Impressions	X	X	X
Mise au point de documents		X	
Conversion de graphiques	X	X	X
Visualisation de graphiques		X	

### 9.10.1.10 Formules

Pour illustrer comment obtenir une formule, voici ce qu'il faut encoder pour obtenir la formule 1.1 reprise ci-dessous :

```
sum from i=0 to n of J sub i
product from i=0 to n of P sub j
integral from 0 to n of f(x)dx
```

$$\sum_{i=0}^n J_i \prod_{j=0}^n P_j \int_0^n f(x)dx \quad [1.1]$$

Autre exemple :

```
left lbrace a over b right rbrace =
left lparen c over d right rparen +
left lbracket e sub i right rbracket
```

$$\left\{ \frac{a}{b} \right\} = \left( \frac{c}{d} \right) + [e_i] \quad [1.2]$$

### 9.10.1.11 Représentation graphique

Le graphique suivant est une représentation en forme de tarte de données issues d'un fichier produit par programme ( voir 5. à la page 99 ). Il a ensuite été converti dans le format requis par DCF pour être inséré dans ce document ( voir 15. à la page 99 ). De la même manière, nous pouvons donc demander d'autres représentations : histogramme, courbe, etc...

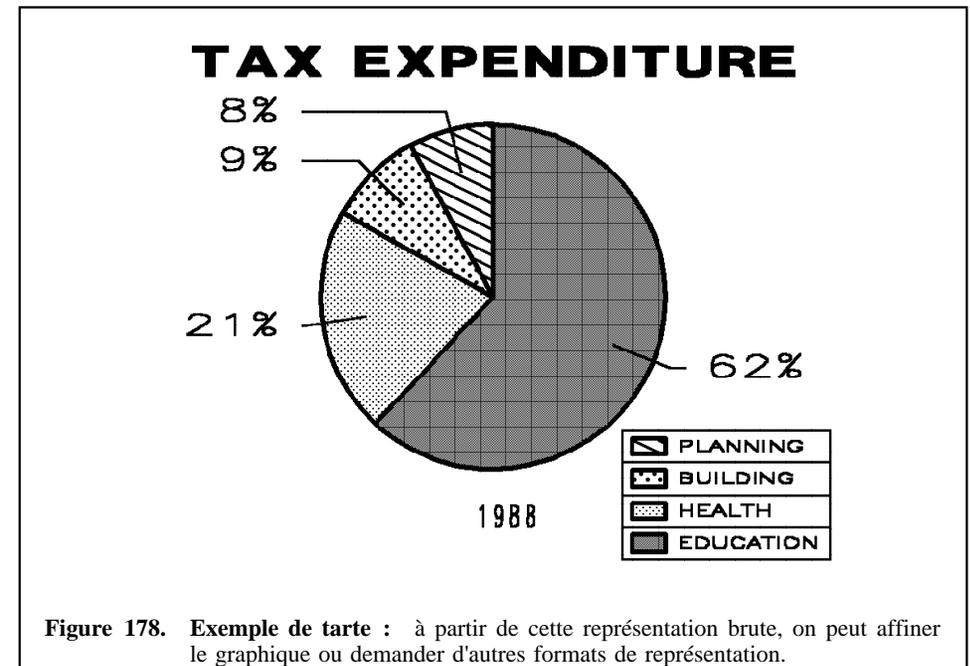


Figure 178. Exemple de tarte : à partir de cette représentation brute, on peut affiner le graphique ou demander d'autres formats de représentation.

### 9.10.1.12 Images digitalisées

Les images suivantes ont été saisies avec un scanner puis réduites aux dimensions requises par le format de la présente brochure.

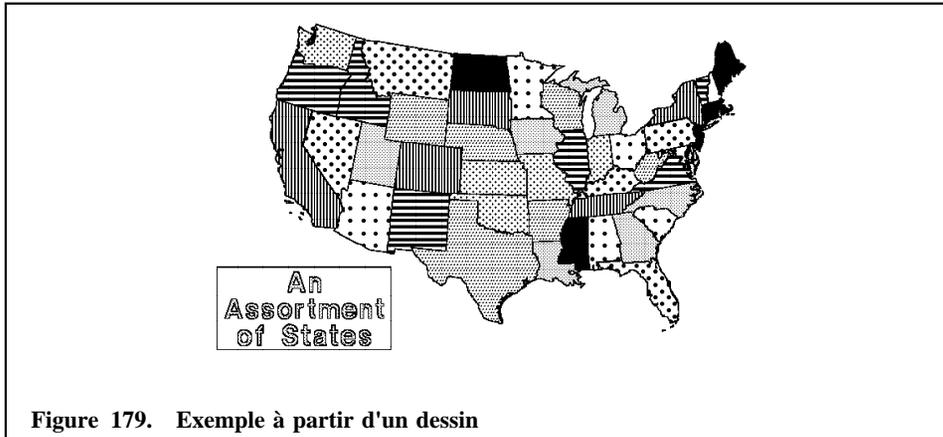


Figure 179. Exemple à partir d'un dessin

### 9.10.2 Les possibilités typographiques

DCF permet d'utiliser différentes familles de symboles appelées FONT et caractérisées chacune par un style (exemple : **Aa** Aa Aa Aa ), par une taille (exemple : Aa Aa), par une intensité (exemple : Aa Aa **Aa** ), par un attribut 'droit', 'italique', 'barré' ou 'contraire' (exemple : Aa Aa Aa Aa ), et par une association code hexadécimal τ symbole imprimé : cette association est appelée un "codepage".

Tous les fonts disponibles en DCF sont décrits dans le "Catalogue des fonts disponibles au SEGI". Cette brochure porte le numéro ZZ01-0108 et est disponible au bureau de documentation du SEGI.

### 9.10.3 Documentation

Lorsqu'on appelle la commande PUBLISH, on dispose, en appuyant sur la touche PF1, d'un "tutorial" qui décrit succinctement les différentes possibilités offertes et qui indique les titres et les numéros des brochures utiles. Ces brochures peuvent être commandées au bureau de documentation du SEGI.

## 9.11 La PUBLIB.

La PUBLIB (Public Library) est constituée d'un ensemble de programmes et sous-programmes de nature utilitaire accompagné d'une documentation enregistrées et d'une procédure d'accès.

La documentation comprend:

- un catalogue enregistré (HELP PUBLIB);
- des notices d'utilisation enregistrées relatives à chaque programme (HELP PUBLIB programme).

Seuls les codes objets des programmes ou sous-programmes sont disponibles.

Procédure:

Procédure d'accès au disque qui contient la PUBLIB:

```
[EXEC] PUBLIB
```

Figure 180. Commande EXEC PUBLIB

Il faut noter que PUBLIB génère les commandes CMS suivantes:

```
GLOBAL TXTLIB PUBLIB PLILIB IBMLIB VSF2LINK VSF2FORT CMSLIB  
GLOBAL LOADLIB
```

qui annule l'effet d'autres commandes GLOBAL qui auraient été exécutée préalablement. Utilisation: après avoir exécuté la commande PUBLIB:

pour un programme: définir les fichiers utilisés par le programme au moyen de commandes FILEDEF appropriées puis exécuter la commande CMS:

```
LOAD programme (START)
```

pour un sous-programme: exécuter le programme qui fait appel à un ou plusieurs sous-programmes de la PUBLIB.

## 9.12 ISPF et ISPF/PDF

### 9.12.1 Introduction.

ISPF et ISPF/PDF sont des produits IBM conçus pour fonctionner sur des terminaux du type 3270 ou compatibles.

Le produit ISPF est une extension du CMS pour faciliter l'implémentation de processus interactifs et offrir une gestion de dialogue à partir d'un programme ou d'une procédure; il permet l'affichage de panneaux et de messages prédéfinis, ainsi que la création, la mise à jour et la consultation de tables de données.

Le produit ISPF/PDF est une aide au développement d'application dans un environnement interactif, particulièrement utile pour tester et contrôler un dialogue ISPF.

## 9.12.2 Utilisation de ISPF

### 9.12.2.1 Généralités

Un dialogue ISPF est activé par une procédure REXX reprenant :

- la définition des fichiers contenant les différents éléments du dialogue (panneaux, messages, tables, fonctions)
- l'appel au produit ISPF

### 9.12.2.2 Définition des fichiers

Ci-dessous, les *filenames* des fichiers définis par l'utilisateur sont notés fn1, fn2, ...; les autres fichiers sont des fichiers du système.

```

/* Profile */
'FILEDEF ISPPROF DISK TABLES MACLIB (PERM)
/* Librairie des panneaux de l'utilisateur */
/* (obligatoire) */
/* concaténée à la librairie du système ISPF */
'FILEDEF ISPPLIB DISK fn1 MACLIB (PERM CONCAT)
'FILEDEF ISPPLIB DISK ISPPLIB MACLIB * (PERM CONCAT)
/* Librairie des messages de l'utilisateur */
/* (optionnelle) */
/* concaténée à la librairie du système ISPF */
'FILEDEF ISPMLIB DISK fn2 MACLIB (PERM CONCAT)
'FILEDEF ISPMLIB DISK ISPMLIB MACLIB * (PERM CONCAT)
/* Librairie des tables utilisées en input */
/* (optionnelle) */
/* concaténée à la librairie du système ISPF */
'FILEDEF ISPTLIB DISK fn3 MACLIB (PERM CONCAT)
'FILEDEF ISPTLIB DISK ISPTLIB MACLIB * (PERM CONCAT)
/* Librairie des tables créées par l'utilisateur */
/* ou utilisées en output */
/* (optionnelle) */
'FILEDEF ISPTABL DISK fn4 MACLIB (PERM CONCAT)
/* Librairie des programmes TEXT de l'utilisateur */
/* et des librairies relatives au langage utilisé */

```

```

/* (optionnelle) */
'FILEDEF ISPXLIB DISK fn5 TXTLIB (PERM CONCAT)
'FILEDEF ISPXLIB DISK PLILIB TXTLIB (PERM CONCAT)
'FILEDEF ISPXLIB DISK IBMLIB TXTLIB (PERM CONCAT)
/* Librairie des programmes LOAD de l'utilisateur */
/* (optionnelle) */
'FILEDEF ISPLLIB DISK fn6 LOADLIB (PERM)

```

### 9.12.2.3 Appel à ISPF

ISPF peut être appelé de trois façons différentes :

soit

```

/* Démarrage de l'application par un programme pgm */
'EXEC ISPSTART PGM(pgm)

```

soit

```

/* Démarrage de l'application par une procédure REXX */
/* proc */
'EXEC ISPSTART CMD(proc)

```

soit

```

/* Démarrage de l'application par un panneau-menu */
/* pan */
'EXEC ISPSTART PANEL(pan)

```

### 9.12.2.4 Exemple

**Définition du problème** Soit la gestion d'un agenda personnel créé dans une table ISPF permanente. : L'application démarre par un panneau-menu qui permet d'activer l'un ou l'autre programme PL/I, qui eux-mêmes affichent des panneaux en relation avec une table ISPF.

#### Procédure d'appel

```

/* Gestion d'un agenda personnel */
'FILEDEF * CLEAR'
'FILEDEF ISPPROF DISK TABLES MACLIB A (PERM)
'FILEDEF ISPPLIB DISK testpdf1 MACLIB A (PERM)
'FILEDEF ISPXLIB DISK testpdf TXTLIB * (PERM)
'FILEDEF ISPXLIB DISK CMSLIB TXTLIB * (PERM CONCAT)

```

```
'FILEDEF ISPXLIB DISK PLILIB  TXTLIB * (PERM CONCAT'
'FILEDEF ISPXLIB DISK IBMLIB  TXTLIB * (PERM CONCAT'
'FILEDEF ISPTLIB DISK testpdf2 MACLIB A (PERM'
'FILEDEF ISPTLIB DISK ISPTLIB  MACLIB * (PERM CONCAT'
'FILEDEF ISPTABL DISK testpdf2 MACLIB A (PERM'
'VMFCLEAR'
'ISPSTART PANEL(panel0)'
'FILEDEF ISPPLIB CLEAR'
'FILEDEF ISPXLIB CLEAR'
'FILEDEF ISPTLIB CLEAR'
'FILEDEF ISPTABL CLEAR'
```

Le contenu des bibliothèques définies par l'utilisateur est le suivant :

- TESTPDF TXTLIB : contient le code objet des programmes TESTPDF1 PLI et TESTPDF2 PLI.
- TESTPDF1 MACLIB : contient les formats d'écran PANEL0, PANEL11, PANAL12, PANEL21, PANEL22.
- TESTPDF2 MACLIB : contient la table créée par le programme TESTPDF1 PLI.

### Panneau menu de sélection (PANEL0)

```
)BODY EXPAND(//) /* PANEL0 */
%/-/ GESTION D'UN AGENDA PERSONNEL /-/
%OPTION ===> _ZCMD +
+
+      %1+ MISE A JOUR
+
+      %2+ AFFICHAGE
+
+      %3+ IMPRESSION
+
+      %X+ FIN DU TRAITEMENT
+
+
+DATE  %&ZDATE
+TIME  %&ZTIME
+
)INIT
)PROC
&ZPF02='PRINT'
VPUT (ZPF02) PROFILE
&ZSEL=TRANS(TRUNC(&ZCMD, '.'))
          1, 'PGM(testpdf1)'
          2, 'PGM(testpdf2)'
          ' ',' '
          X, 'EXIT'
          *, '?' )
)END
```

## Programme PL/I de création/mise à jour (TESTPDF1)

```
TESTPDF1 : PROC OPTIONS(MAIN);
DCL ISPLINK ENTRY EXTERNAL OPTIONS(ASM INTER RETCODE),
    PLIRETV BUILTIN;
DCL (JJ,MM,AA,HR,MN) CHAR(2),
    ACTIV CHAR(20),
    L02 FIXED BIN(31,0) INIT(2),
    L08 FIXED BIN(31,0) INIT(8),
    L20 FIXED BIN(31,0) INIT(20),
    RC FIXED BIN(31) INIT(0),
    RC_EXIST FIXED BIN(31) INIT(0);

DCL ZPF03 CHAR(8) INIT('END'),
    ZPF02 CHAR(8) INIT('PRINT');

CALL ISPLINK('VDEFINE','ZPF02 ',ZPF02,'CHAR',L08);
CALL ISPLINK('VDEFINE','ZPF03 ',ZPF03,'CHAR',L08);
CALL ISPLINK('VPUT','(ZPF02 ZPF03)','PROFILE');
CALL ISPLINK('VDELETE','(ZPF02 ZPF03)');

/* OU ... VGET ... VREPLACE ... VPUT */

CALL ISPLINK('VDEFINE','JJ ',JJ,'CHAR',L02);
/* OU
CALL ISPLINK('VDEFINE','(JJ)',JJ,'CHAR',L02); */
CALL ISPLINK('VDEFINE','MM ',MM,'CHAR',L02);
CALL ISPLINK('VDEFINE','AA ',AA,'CHAR',L02);
CALL ISPLINK('VDEFINE','HR ',HR,'CHAR',L02);
CALL ISPLINK('VDEFINE','MN ',MN,'CHAR',L02);
CALL ISPLINK('VDEFINE','ACTIV ',ACTIV,'CHAR',L20);
```

```
CALL ISPLINK('TBOPEN','TESTPDF2');
IF PLIRETV()= 8 THEN DO; /* TABLE INEXISTANTE */
    CALL ISPLINK('TBCREATE','TESTPDF2',
        '(JJ,MM,AA,HR,MN)', 'ACTIV');
    CALL ISPLINK('TBSORT','TESTPDF2',
        '(JJ,C,A,MM,C,A,AA,C,A,HR,C,A,MN,C,A)');
    END;

DO WHILE(RC=0);

    CALL ISPLINK('DISPLAY','PANEL11 ');

    CALL ISPLINK('TBTOP','TESTPDF2');
    CALL ISPLINK('TBSCAN','TESTPDF2','(JJ,MM,AA,HR,MN)');
    RC_EXIST=PLIRETV();
    IF RC_EXIST=8 THEN ACTIV=' '; /*LIGNE INEXISTANTE*/

    CALL ISPLINK('DISPLAY','PANEL12 ');
    RC=PLIRETV();
    IF ACTIV^=' ' THEN
        CALL ISPLINK('TBMOD','TESTPDF2',' ','ORDER ');
        ELSE IF RC_EXIST=0 THEN
            CALL ISPLINK('TBDELETE','TESTPDF2');
    ACTIV='BIDON';    CALL ISPLINK('VPUT','(ACTIV)','PROFILE');
END;
CALL ISPLINK('TBCLOSE','TESTPDF2');
CALL ISPLINK('VDELETE','(JJ,MM,AA,HR,MN,ACTIV)');
/* OU
CALL ISPLINK('VRESET');
*/
END TESTPDF1;
```

## Panneaux de création/mise à jour (PANEL11 et PANEL12)

```
)ATTR
! TYPE(TEXT) INTENS(LOW) SKIP(ON)
)BODY EXPAND($$) /* PANEL11 */
%$-$ GESTION D'UN AGENDA PERSONNEL $-$
%COMMAND ==> _ZCMD
+
+     ENTREZ LA DATE ET L'HEURE :
+     DATE   ==> _JJ!/_MM!/_AA!       (JJ/MM/AA)
+     HEURE  ==> _HR!:_MN+           (HH:MM)
+
+DATE %&ZDATE
+TIME %&ZTIME
)INIT
&ZCMD=' '
.CURSOR=JJ
)END

)ATTR
! TYPE(TEXT) INTENS(LOW) SKIP(ON)
# TYPE(OUTPUT)
)BODY EXPAND($$) /* PANEL12 */
%$-$ GESTION D'UN AGENDA PERSONNEL $-$
%COMMAND ==> _ZCMD
+
+     ENTREZ LA DESCRIPTION DE L'ACTIVITE :
+     DATE   ==> #JJ+/#MM+/#AA+       (JJ/MM/AA)
+     HEURE  ==> #HR+:#MN+           (HH:MM)
+     ACTIVITE ==> _ACTIV+
+
+DATE %&ZDATE
+TIME %&ZTIME
)INIT
&ZCMD=' '
)END
```

## Programme PL/I d'affichage (TESTPDF2)

```
TESTPDF2 : PROC OPTIONS(MAIN);
DCL ISPLINK ENTRY EXTERNAL OPTIONS(ASM INTER RETCODE),
     PLIRETV BUILTIN;
DCL (JJ,MM,AA,HR,MN) CHAR(2),
     (JS,MS,AS) CHAR(2),
     ACTIV CHAR(20),
     L02 FIXED BIN(31,0) INIT(2),
     L05 FIXED BIN(31,0) INIT(05);

CALL ISPLINK('VDEFINE','JS ','JS','CHAR',L02);
CALL ISPLINK('VDEFINE','MS ','MS','CHAR',L02);
CALL ISPLINK('VDEFINE','AS ','AS','CHAR',L02);
CALL ISPLINK('VDEFINE','JJ ','JJ','CHAR',L02);
CALL ISPLINK('VDEFINE','MM ','MM','CHAR',L02);
CALL ISPLINK('VDEFINE','AA ','AA','CHAR',L02);
CALL ISPLINK('VDEFINE','HR ','HR','CHAR',L02);
CALL ISPLINK('VDEFINE','MN ','MN','CHAR',L02);
CALL ISPLINK('VDEFINE','ACTIV ','ACTIV','CHAR',L05);

CALL ISPLINK('TBOPEN','TESTPDF2');
IF PLIRETV()= 8 THEN ; /* TABLE INEXISTANTE */
ELSE DO;

CALL ISPLINK('DISPLAY','PANEL21 ');
CALL ISPLINK('TBVCLEAR','TESTPDF2');
JJ=JS;MM=MS;AA=AS;
CALL ISPLINK('TBSARG ','TESTPDF2',' ','NEXT',
             '(JJ,EQ,MM,EQ,AA,EQ)');
IF PLIRETV()=8 THEN; /* ARGUMENTS DE RECHERCHE
NON INITIALISES */
ELSE
CALL ISPLINK('TBDISPL','TESTPDF2','PANEL22 ');
CALL ISPLINK('TBCLOSE','TESTPDF2');

END;

CALL ISPLINK('VDELETE','(JJ,MM,AA,HR,MN,ACTIV)');
CALL ISPLINK('VDELETE','(JS,MS,AS)');
END TESTPDF2;
```

## Panneaux d'affichage (PANEL21 et PANEL22)

```
)ATTR
! TYPE(TEXT) INTENS(LOW) SKIP(ON)
)BODY EXPAND($$) /* PANEL21 */
%$-$ GESTION D'UN AGENDA PERSONNEL $-$
%COMMAND ==> _ZCMD
+
+     ENTREZ LA DATE DES ACTIVITES A AFFICHER :
+
+     DATE ==> _JS!/_MS!/_AS+      (JJ/MM/AA)
+DATE %%ZDATE
+TIME %%ZTIME
)INIT
&ZCMD=' '
)END

)ATTR
! TYPE(TEXT) INTENS(LOW) SKIP(ON)
# TYPE(OUTPUT)
)BODY EXPAND($$) /* PANEL22 */
%$-$ GESTION D'UN AGENDA PERSONNEL $-$
%COMMAND ==> _ZCMD          %SCROLL ==>_AMT +
+
+     LISTE DES ACTIVITES A LA DATE DU :
+
+     #JS!/#MS!/#AS+      (JJ/MM/AA)
+
+DATE %%ZDATE
+TIME %%ZTIME
% HEURE      ACTIVITE
%
)MODEL ROWS(SCAN)
#HR+:#MN      +#ACTIV+
+$-$
+
)INIT
&ZCMD=' '
&AMT=HALF
)END
```

## 9.12.3 Utilisation de ISPF/PDF

Le produit ISPF/PDF est appelé par la procédure PDF. Il offre d'une part différents services au niveau du développement d'applications, -fonctions de "browse", d'"edit", de compilation et de gestion de bibliothèques ou de fichiers-, et d'autre part des outils de test d'un dialogue ISPF.

Signalons que les différents services supportés par le produit ISPF/PDF peuvent être appelés dans un dialogue ISPF.

## 9.13 Logiciel de calcul symbolique: *REDUCE*.

Selon le propos de ses auteurs:

REDUCE is a program designed for general algebraic computations of interest to mathematicians, physicists and engineers

Its capabilities include:

- expansion and ordering of polynomials and rational functions,
- substitutions and pattern matching in a wide variety of forms,
- automatic and user controlled simplification of expressions,
- calculations with symbolic matrices,
- arbitrary precision integer and real arithmetic,
- facilities for defining new functions and extending program syntax,
- analytic differentiation and integration,
- factorization of polynomials,
- Dirac matrix calculations of interest to high energy physicists,
- a complete LISP language for symbolic calculations, in which REDUCE itself is written

REDUCE est distribué par l'Université de Salt Lake City, Utah, USA, et est largement diffusé dans le monde.

### 9.13.1 Utilisation.

La procédure à appeler pour entrer dans l'environnement REDUCE est:

```
REDUCE [(options)]
```

Figure 181. Commande EXEC REDUCE

Les options de cette procédure sont exposées en réponse à la commande **HELP SEGI REDUCE** ; elles permettent de préciser:

l'origine des commandes (console, fichier disque),

la destination des résultats (console, imprimante, fichier disque),

la possibilité de conservation d'une trace console,

le système de programmation à charger en mémoire:

REDUCE lui-même (par défaut),

RLISP, mode symbolique de REDUCE,

SLISP, langage symbolique STANDARD LISP.

RESTART, point de reprise assuré au cours d'une session antérieure par l'instruction LISP: (CHKPOINT (QUOTE RESTART))

**Remarque :** REDUCE comprend RLISP, qui lui-même comprend SLISP.

un jeu de fonctions précompilées à charger en même temps que le système.

Les instructions à exécuter pour retourner en CMS sont:

BYE; ou QUIT; en venant de REDUCE ou de RLISP

(QUIT) en venant de SLISP

**Remarque :** il est toujours nécessaire d'assurer explicitement le retour à CMS par l'instruction adéquate, même lorsque les instructions sont contenues dans un fichier.

## 10.0 Aspects administratifs et comptables

### 10.1 Structure et fonctionnement des comptes

Le double principe de l'imputation des coûts au service utilisateur et de la provision préalable à toute consommation implique la tenue de comptes courants dans la comptabilité du SEGI. D'autre part, comme on l'a vu précédemment, l'accès au superordinateur en VM requiert l'emploi d'une machine virtuelle (identifiée par un "userid") et la disponibilité d'un certain espace sur disque (minidisque).

Il en résulte que, pour chaque service utilisateur, une structure d'informations à 4 niveaux doit être enregistrée par le SEGI préalablement à toute utilisation du système VM:

1. un compte de service,
2. un (ou plusieurs) compte(s) de travail,
3. une (ou plusieurs) machine(s) virtuelle(s),
4. un (ou plusieurs) minidisque(s).

#### 10.1.1 Le compte de service

Tout service utilisateur se voit attribuer par le SEGI un numéro d'identification de 4 caractères alphanumériques, qui lui est propre, ainsi qu'un "compte de service" portant ce numéro.

Ce compte est crédité des ressources que le chef de service peut affecter à l'emploi de l'ordinateur (crédits obtenus des Autorités, ou fonds propres versés au SEGI).

#### 10.1.2 Les comptes de travail

Le chef de service peut distribuer son budget entre les différents travaux (tp, tfe, recherches...) qu'il souhaite faire réaliser, en ouvrant des "comptes de travail" particuliers où les prestations seront imputées; l'ouverture de tels comptes est d'ailleurs indispensable aux traitements en ordinateur.

Ces comptes sont approvisionnés par prélèvement sur le compte de service, et sont débités des consommations des machines virtuelles (CPU,I/O,...) et des minidisques.

En VM, les comptes de travail sont identifiés par un "numéro" de 8 caractères, dont les 4 premiers sont ceux du numéro de service et dont les suivants ont pour signification:

- le 5ème correspond à l'année d'ouverture (A pour 81,...P pour 96)
- les 3 derniers à un numéro de séquence dans l'année

### 10.1.3 Les machines virtuelles

Elles sont identifiées par un "userid" de maximum 8 caractères et leurs consommations sont imputées sur un compte de travail déterminé; notons à cet égard qu'une machine peut, lors de son ouverture, être associée à un compte de travail ou à plusieurs et que, dans ce dernier cas, ses consommations sont imputées sur le compte choisi à un moment donné par l'utilisateur (via la commande "CP SET ACCOUNT"); d'autre part, plusieurs machines peuvent être associées à un même compte de travail.

### 10.1.4 Les minidisques

Ils sont liés directement aux machines virtuelles, et l'espace qu'ils occupent est comptabilisé sur un compte de travail déterminé (choisi par l'utilisateur lors de l'ouverture de la machine); ce compte peut très bien être différent de celui sur lequel sont imputées les consommations de la machine elle-même. Il est par ailleurs possible d'imputer sur un même compte l'espace réservé par plusieurs machines.

Enfin, pour tout compte ainsi lié à des minidisques, le SEGI indique à l'utilisateur l'espace maximum qu'il est autorisé à réserver (tout dépassement de ce "contrat-disque" étant facturé à un tarif de dissuasion, 25 fois plus coûteux que le taux normal); l'absence de dépassement peut être vérifiée aisément et à tout moment par la commande "SPACE LIMIT \*", ou encore la commande "COSTDISK" (qui indique en plus le coût journalier).

## 10.2 Ouverture des comptes et des machines virtuelles

Compte tenu de ce qui précède, les formalités préalables à l'utilisation du système VM par un service universitaire sont les suivantes (s'adresser à Monsieur PETIT) :

1. Au départ, le service doit disposer dans la comptabilité du SEGI d'un budget-calcul (crédits obtenus ou versements de fonds); son montant est porté au crédit du "compte de service".
2. Il fera ouvrir et approvisionner un ou plusieurs comptes de travail; les provisions de ces comptes devront être suffisantes pour couvrir les consommations: avant l'exécution d'un travail (job ou session), le disponible du compte employé est en effet contrôlé.

Modalités pratiques:

Renvoi du formulaire "Demande d'ouverture d'un compte de travail" fourni en annexe (cfr «formulaire 1» à la page 117), ou utilisation de la commande CREDIT (voir infra) par le responsable du service ou son délégué.

3. Il demandera la création d'une ou plusieurs machines virtuelles comprenant chacune, au départ, un minidisque d'adresse 191 de maximum 3 cylindres de disque 3390; remarquons ici qu'un minidisque comporte toujours un nombre entier de cylindres réels sur les disques 3390 (où 1 cyl. = 15 pistes).

Modalités pratiques:

Renvoi du formulaire "Demande de création d'une machine virtuelle" figurant en annexe (cfr «formulaire 2» à la page 117).

4. Si nécessaire, il pourra demander par la suite un accroissement d'espace-disque, soit par l'extension d'un minidisque existant, soit par la création d'un minidisque supplémentaire.

Modalités pratiques:

Renvoi du formulaire "Demande d'espace-disque supplémentaire" figurant en annexe (cfr «formulaire 3» à la page 118).

## ***10.3 Imputation des consommations sur les comptes***

Dès la fin d'un travail, qu'il s'agisse d'une session interactive ou d'un travail batch, ses consommations sont immédiatement valorisées au tarif en vigueur et imputées sur le compte de travail utilisé.

Les coûts relatifs à la réservation de minidisques permanents sont calculés chaque jour (au prorata des pistes réservées), et imputés sur le compte de travail concerné; quant aux redevances forfaitaires d'utilisation des machines virtuelles, elles sont imputées sur les comptes de travail une fois par mois.

**N.B.** Les coûts relatifs aux tracés graphiques sur table CALCOMP sont imputés une fois par mois et débités du compte de service (et non des comptes de travail).

## ***10.4 Notification des coûts aux utilisateurs***

En fin de session ou de job batch, l'utilisateur obtient le détail de ses consommations (temps de connexion, temps CPU, nombre d'IO, ...) ainsi qu'une indication des coûts imputés et de la provision restant disponible sur le compte de travail concerné.

L'utilisateur a également la possibilité, en mode batch ou interactif, de connaître à tout moment l'état des consommations en cours, et le solde prévisionnel de son compte de

travail, par la commande COST. Le coût journalier de l'espace-disque utilisé peut quant à lui être obtenu en mode interactif par la commande COSTDISK.

Par ailleurs, chaque service universitaire qui le souhaite peut recevoir mensuellement, par courrier, un extrait de compte présentant la situation globale de son compte de service et de ses comptes de travail, pour l'année civile en cours. Ajoutons encore que cette situation globale peut être visualisée en permanence (par une personne autorisée) grâce à la commande CREDIT (voir ci-après).

Enfin, une commande, dénommée ACCOFF, permet aux services qui le souhaitent, d'obtenir sur listing tout ou partie du détail de leurs consommations d'un mois donné, celles-ci étant enregistrées sur disques et accessibles pendant la seconde quinzaine du mois suivant.

Cette commande présente les caractéristiques suivantes :

- elle est conversationnelle, et permet à l'utilisateur de sélectionner les niveaux d'informations voulus, sur base de différents critères; la sélection une fois définie, la commande déclenche l'exécution d'un job "batch" qui renvoie les résultats dans le "reader" de la machine de l'utilisateur
- elle exige l'introduction du mot de passe associé à la commande CREDIT; il s'agit du mot de passe "actif" au moment du passage mensuel des programmes de facturation et de la constitution des fichiers-détails sur disques (celle-ci a lieu dans la 1ère quinzaine du mois qui suit le mois facturé). La commande n'est accessible en outre qu'aux seules machines virtuelles d'utilisateurs expressément autorisés par le SEGI (s'adresser à J-M PETIT).

## ***10.5 Gestion des comptes par la commande CREDIT***

Comme indiqué plus haut, l'emploi du système VM par les services universitaires exige toujours l'ouverture préalable d'un ou plusieurs comptes de travail et leur approvisionnement régulier par prélèvement sur le "compte de service".

La simplification de ces approvisionnements et des diverses opérations de gestion des comptes de travail est l'objet même de la commande CREDIT; elle permet notamment au responsable du service (ou à son délégué):

- de connaître la situation de son compte de service, et celle de chacun des comptes de travail (provisions, consommations, solde); voir sous-commandes SOLDE et LIST;
- de modifier la provision allouée à un compte de travail (modification pouvant être négative); voir sous-commande PROVISION;
- d'afficher toutes les informations caractérisant un compte de travail (par ex.: crédit/débit, dates de validité, type du travail, titulaire, tél,...) et d'en modifier certaines; voir sous-commande AFFICHE;

- d'ouvrir un nouveau compte de travail (à condition qu'il en existe déjà un pour le système utilisé) ou de clôturer un compte dont l'emploi n'est plus souhaité; voir sous-commandes AJOUTE et CLOTURE.

L'emploi de CREDIT est en outre protégé par un mot de passe spécial (différent de celui du LOGON).

Pour pouvoir utiliser la commande, le chef de service ou son délégué contactera J.M. Petit, ou remplira le formulaire intitulé "Demande d'accès à la commande CREDIT" (Voir «formulaire 4» à la page 118) et le transmettra au SEGI. Les modalités d'emploi de CREDIT sont quant à elles détaillées dans un HELP interne à la commande.

## ***10.6 Services particuliers facturés aux utilisateurs***

Des facilités spéciales, comme l'utilisation du réseau DCS (X.25) de Belgacom et l'accès à la petite imprimante LASER 3812, sont disponibles moyennant un financement par transfert de crédit ordinaire ou de compte Patrimoine (et non par des crédits-calcul). Toute personne désireuse de recourir à l'un ou l'autre de ces services fera préalablement remplir et signer par son chef de service une demande d'accès (Voir «formulaire 5» à la page 119, et «formulaire 6» à la page 119); l'autorisation d'accès sera liée au "userid" désigné par le demandeur.



## 11.0 Bibliographie

### 11.1 IBM

- gc24-5550** VM/ESA General Information
- sc24-5458** VM/ESA CMS Primer
- sc24-5519** VM/ESA CP Command and Utility Reference
- sc24-5461** VM/ESA CMS Command Reference
- sc24-5460** VM/ESA CMS User's Guide
- sc24-5463** VM/ESA XEDIT User's Guide
- sc24-5464** VM/ESA XEDIT Command and Macro Reference
- sc24-5465** VM/ESA REXX/VM User's Guide
- sc24-5466** VM/ESA REXX/VM Reference
- sc24-5463** VM/ESA CMS Pipelines User's Guide
- sc24-5464** VM/ESA CMS Pipelines Reference
- sc24-5450** VM/ESA CMS Application Development Guide
- sc24-5529** VM/ESA System Messages and Codes
- sc19-6206** IBM VM/SP Terminal User's Guide
- sh20-9227** APL2 Programming Language
- sh20-9216** APL2 Programming Guide
- gc26-4037** Assembler H version 2, Language Reference
- sc26-4036** Assembler H version 2, Programming Guide
- gc26-4047** VS COBOL II Language Reference
- sc26-4045** VS COBOL II Application Programming Guide
- sc26-4214** VS COBOL II Application Programming Supplement for CMS Users
- sc26-4049** VS COBOL II Debugging
- sc26-4301** COBOL Report Writer Precompiler Programmer's Guide
- sc26-4320** VS Pascal Language Reference
- sc26-4319** VS Pascal Application Programming Guide
- sc26-4308** OS PL/I version 2, Programming Language Reference
- sc26-4307** OS PL/I version 2, Programming Guide
- sc26-4310** OS PL/I version 2, Programming using PLITEST
- sc26-4221** VS FORTRAN Language and Library Reference
- sc26-4222** VS FORTRAN Programming Guide
- sc26-4223** VS FORTRAN Interactive Debug, Guide and Reference
- sc23-0526** Engineering and Scientific Subroutine Library, Guide and Reference
- sc26-4668** VAST-2 for VS FORTRAN, User's Guide
- sc09-1308** Common Programming Interface C Reference
- sc09-1384** IBM C/370 Programming Guide
- sc34-4270** Interactive System Productivity Facility (ISPF) and ISPF/Program Development Facility (ISPF/PDF), ISPF and ISPF/PDF Primer
- sc34-4306** Interactive System Productivity Facility / Program Development Facility, ISPF/PDF Guide
- sc34-4307** Interactive System Productivity Facility / Program Development Facility, ISPF/PDF Services
- sc34-4308** Interactive System Productivity Facility / Program Development Facility, Library Management Facility
- sc34-4273** Interactive System Productivity Facility (ISPF) ISPF Dialog Management Guide and Reference
- sc34-4265** Interactive System Productivity Facility (ISPF) ISPF Dialog Management Examples
- sc34-4267** Interactive System Productivity Facility (ISPF) Dialog Tag Language Guide and Reference
- sh35-0069** Document Composition Facility: SCRIPT/VS Text Programmer's Guide
- sh35-0070** Document Composition Facility: SCRIPT/VS Language Reference
- sx26-3723** Document Composition Facility: SCRIPT/VS Text Programmer's Quick Reference

- sh35-0048** Document Composition Facility Messages
- gh20-9158** Document Composition Facility and Document Library Facility General Information
- sh20-9186** Document Composition Facility: Generalized Markup Language : Starter Set User's Guide
- sh20-9187** Document Composition Facility: Generalized Markup Language : Starter Set Reference
- sh35-0050** Document Composition Facility: Generalized Markup Language Starter Set Implementation Guide
- sh20-9188** Document Composition Facility: Generalized Markup Language Concepts and Design Guide
- sx26-3719** Document Composition Facility: Generalized Markup Language Quick Reference
- sh20-6453** Script Mathematical Formula Formatter User's Guide
- gh12-5822** IBM Search Manager/370, General Information
- sh12-5903** IBM Search Manager/370, Administration Guide for VM
- sh12-5825** IBM Search Manager/370, User's Guide for MVS and VM
- gh09-8044** SQL/Data System Concepts and Facilities
- GC33-0319** Graphical Data Display Manager General Information, IBM
- GC33-0125** Graphics Access Method, System Product, General Information, IBM

## **11.2 SEGI**

- zz01-0099** La Table traçante CALCOMP, Description, Utilisation, Logiciel de base
- zz01-0102** La procédure S.A.S. DISSIM
- zz01-0105** Accès au réseau X.25 par DCS
- zz01-0108** Catalogue des FONTS disponibles au SEGI
- zz01-0109** SAS/GRAPH au SEGI
- zz01-0110** DISSPLA au SEGI
- zz01-0111** Optimisation, Vectorisation et parallélisation des programmes FORTRAN
- zz01-0113** Guide pratique d'utilisation du logiciel FTPSEGI

- zz01-0114** Manuel de référence du logiciel FTPSEGI
- zz01-0117** Guide d'Introduction à l'Inter réseau ULg

## **11.3 SAS**

- SAS Language Reference.
- SAS Procedures Guide.
- SAS Language and Procedures: Usage.
- SAS Language and Procedures: Syntax.
- SAS Companion for the CMS Environment.
- SAS Guide to the SQL Procedure.
- SAS Technical Report P-222, Changes and Enhancements to Base SAS Software.
- SAS/STAT User's Guide (2 volumes).
- SAS Technical Report P-229, SAS/STAT Software: Changes and Enhancements.
- SAS/ETS User's Guide.
- SAS/ETS Software: Applications Guide.
- SAS/OR User's Guide.
- SAS/GRAPH Software: Introduction.
- SAS/GRAPH Software: Reference (2 volumes).
- SAS/GRAPH Software: Usage.
- SAS/GRAPH Software: Syntax.
- SAS/GRAPH Software: Graphics Editor.
- SAS Technical Report P-125, SAS/GRAPH Software: Changes and Enhancements.
- SAS/FSP Software: Usage and Reference.
- Getting started with SAS/CALC Spreadsheet Applications.
- SAS/CALC Software: Usage and Reference.
- SAS/IML Software: Usage and Reference.

----- SAS/CONNECT Software: Usage and Reference.

## ***11.4 DISSPLA***

----- CA-DISSPLA User's Manual, Computer Associates.

----- DISSPLA Codebook User's Manual, Computer Associates.

----- CA-DISSPLA First Facts, Computer Associates.

----- CA-DISSPLA Pocket Guide, Computer Associates.

## ***11.5 Divers***

----- The Generalised Interactive Modelling System, Manual (GLIM).

----- BMDP Statistical Software, University of California (2 volumes).

----- NAG FORTRAN LIBRARY Manual (10 volumes).

----- NAG FORTRAN Mini Manual.

----- CASORT/CMS Users Guide VM, Computer Associates.

----- REDUCE USER'S MANUAL, University of Utah.

----- STANDARD LISP REPORT, University of Utah.

----- STANDARD LISP ON IBM SYSTEM, University of Utah.



A.1 formulaire 1

SEGI-ULG Compte(attribué par SEGI) > . . . . . <

DEMANDE D'OUVERTURE D'UN COMPTE DE TRAVAIL

No DE SERVICE AU SEGI > . . . . . <
TITRE > . . . . . <
BREF EXPOSE > . . . . . <
> . . . . . <
TITULAIRE > . . . . . <
ADRESSE > . . . . . <
BAT > . . . . . < TEL > . . / . . . . . < EXT > . . <
PROVISION INITIALE ALLOUEE AU COMPTE > . . . . . <
CATEGORIE DU TRAVAIL (cochez par une x)
-cours/trav. pratiques > < -t.f.e./mémoire lic. > <
-thèse doct./agrégat. > < -recherche non subsid > <
-recherche subsidiée par Fondations > <
-recherche subsidiée par d'autres organismes > <
-tâches de gestion > <
-prestations retribuées
-recherches > < -exploitation > <
LE CAS ECHEANT, INDIQUEZ
-les limites de validité (JJMAA) du > . . . . . <
au > . . . . . <
-pour NE PAS voir imprimer en fin de travail
(cochez par une x) -les consommations > <
-les coûts > <
-le crédit/débit > <
DATE > . . . . . <
SIGNATURE DU RESPONSABLE > . . . . . <

RESERVE AU SEGI

Cette demande a été enregistrée le > . . . . . <
Remarque éventuelle > . . . . . <
Signature > . . . . . <

A.2 formulaire 2

SEGI-ULG

DEMANDE DE CREATION D'UNE MACHINE VIRTUELLE (OU DE MODIFICATION DE SES CARACTERISTIQUES)

No DE SERVICE AU SEGI > . . . . . <
UTILISATEUR DE LA MACHINE > . . . . . <
FONCTION > . . . . . <
ADRESSE > . . . . . <
BAT > . . . . . < TEL > . . / . . . . . < EXT > . . <
Userid de la machine > . . . . . <(max.8 caract.)
Mot de passe machine > . . . . . <(max.8 caract.)
Retour des listings DEST= > . . . < CASE= > . . . <
Taille du mini-disque 191 > < max. 3 cyl.disk 3390
Mode d'accès au mini-disque 191 > < (par défaut M)
Autorisations particulières ACF2
> . . . . . <
> . . . . . <
> . . . . . <
No du compte associé au mini-disque > . . . . . <
No(s) du(des) compte(s) associé(s) à la machine (max.
8 comptes) > . . . . . <
Userid(s) de machine(s) dépendante(s) > . . . . . <
> . . . . . <
DATE > . . . . . <
SIGNATURE DU RESPONSABLE > . . . . . <

RESERVE AU SEGI

La présente demande a été enregistrée le > . . . . . <
Remarque éventuelle > . . . . . <
Signature > . . . . . <

### A.3 formulaire 3

SEGI-ULG

DEMANDE DE RESERVATION D'ESPACE-DISQUE  
SUPPLEMENTAIRE POUR UNE MACHINE VIRTUELLE

-----  
No DE SERVICE AU SEGI > . . . . . <  
UTILISATEUR DE LA MACHINE > . . . . . <  
FONCTION > . . . . . <  
ADRESSE > . . . . . <  
BAT > . . . . . < TEL > . . / . . . . . < EXT > . . <  
Userid de la machine > . . . . . <  
Espace supplémentaire désiré > . . . . . < cylindres 3390  
- à ajouter au mini-disque d'adresse > . . . <  
    existant déjà  
ou  
- à attribuer à un nouveau mini d'adresse > . . . <  
Dans le second cas, indiquez  
  . le mode d'accès de ce mini > . . . < (par défaut, M)  
  . les éventuelles autorisations particulières ACF2  
> . . . . . <  
> . . . . . <  
> . . . . . <  
No du compte associé au mini-disque > . . . . . <  
DATE > . . . . . <  
SIGNATURE DU RESPONSABLE > . . . . . <  
-----

RESERVE AU SEGI

-----  
La présente demande a été enregistrée le > . . . . . <  
Remarque éventuelle > . . . . . <  
Signature > . . . . . <

### A.4 formulaire 4

SEGI-ULG      DEMANDE D'ACCES A LA COMMANDE CREDIT

-----  
No DE SERVICE AU SEGI > . . . . . <  
INTITULE DU SERVICE > . . . . . <  
ADRESSE > . . . . . <  
BAT > . . . . . < TEL > . . / . . . . . < EXT > . . <

Je soussigné, chef du service sus-mentionné, souhaite  
que mes comptes de travail ouverts au SEGI puissent  
être mis à jour par terminaux,  
    j'effectuerai ces opérations personnellement;  
(\* )  
    je délègue à cet effet Mr, Mme >.....<

Je note que ces opérations, exécutées par moi-même  
ou par mon délégué, se feront sans intervention  
du SEGI, celui-ci étant dégagé de toute obligation  
de vérification. Notamment, au cas où l'entièreté du  
disponible du service serait affectée aux comptes de  
travail, les graphiques CALCOMP effectués par  
mon service pourraient faire l'objet d'une facturation  
distincte. La gestion du mot de passe ne relevant pas  
du SEGI, celui-ci n'assumera aucune responsabilité  
au cas où il viendrait à être divulgué.  
MOT DE PASSE D'ACCES A "CREDIT" > . . . . . <  
DATE > . . . . . <  
SIGNATURE DU RESPONSABLE > . . . . . <  
(\* ) Biffer la mention inutile

RESERVE AU SEGI

-----  
Cette demande a été enregistrée le > . . . . . <  
Remarque éventuelle . . . . . <  
Signature > . . . . . <

## A.5 formulaire 5

SEGI-ULG DEMANDE D'ACCES AU RESEAU DCS  
VIA LES INSTALLATIONS DU SEGI

-----

No DU SERVICE AU SEGI > . . . . . <  
INTITULE DU SERVICE > . . . . . <  
ADRESSE > . . . . . <  
BAT > . . . . . < TEL > . . / . . . . . < EXT > . . <  
Je soussigné, chef du service sus-mentionné, souhaite  
que celui-ci puisse accéder au réseau DCS de Belgacom  
à partir de terminaux reliés au SEGI.  
L'autorisation d'appeler le réseau sera attribuée  
au "userid" suivant:  
USERID VM AUTORISE POUR DCS > . . . . . <  
PRINCIPAL UTILISATEUR DU USERID > . . . . . <

Je note que les communications émises par le "userid"  
indiqué ci-dessus seront périodiquement refacturées à  
mon service par le SEGI, au tarif appliqué par Belgacom;  
ces frais seront nécessairement financés par un  
transfert de crédit ordinaire ou de compte Patrimoine  
(et non par des crédits-calcul). D'autre part, la  
gestion du mot de passe associé au "userid" ne  
relevant pas du SEGI, celui-ci n'assumera aucune  
responsabilité au cas où il viendrait à être divulgué.

DATE > . . . . . <  
SIGNATURE DU RESPONSABLE > . . . . . <

-----

RESERVE AU SEGI

-----

Cette demande a été enregistrée le > . . . . . <  
Remarque éventuelle > . . . . . <  
Signature > . . . . . <

## A.6 formulaire 6

SEGI-ULG DEMANDE D'ACCES A  
L'IMPRIMANTE LASER 3812

-----

No DU SERVICE AU SEGI > . . . . . <  
INTITULE DU SERVICE > . . . . . <  
ADRESSE > . . . . . <  
BAT > . . . . . < TEL > . . / . . . . . < EXT > . . <  
Je soussigné, chef du service sus-mentionné,  
souhaite que celui-ci puisse utiliser l'imprimante  
laser IBM 3812. L'autorisation d'emploi sera  
attribuée au "userid" suivant:  
USERID VM > . . . . . <  
PRINCIPAL UTILISATEUR DU USERID > . . . . . <

Je note que les impressions transmises par le  
"userid" indiqué ci-dessus seront périodiquement  
facturées à mon service par le SEGI; ces  
frais seront nécessairement financés par un  
transfert de crédit ordinaire ou de compte  
Patrimoine (et non par des crédits-calcul).  
D'autre part, la gestion du mot de passe  
associé au "userid" ne relevant pas du SEGI,  
celui-ci n'assumera aucune responsabilité au  
cas où il viendrait à être divulgué.

DATE > . . . . . <  
SIGNATURE DU RESPONSABLE > . . . . . <

-----

RESERVE AU SEGI

-----

La présente demande a été enregistrée le > . . . . . <  
Remarque éventuelle > . . . . . <  
Signature > . . . . . <



## Annexe B. Les jeux de caractères employés au SEGI

Depuis quelques années, les organismes internationaux de normalisation ont défini une norme connue sous l'appellation ISO 8859/1. Cette norme énumère 191 graphismes de caractères qui couvrent l'ensemble des particularités (accentuation notamment) propres aux différentes langues qui utilisent l'écriture "latine". Cette norme s'applique donc pratiquement à toute l'Europe de l'Ouest, aux Etats Unis, à l'Amérique en général...

Les caractères ISO 8859/1 sont associés à des codes différents suivant l'environnement informatique dans lequel on travaille. C'est ainsi que dans le monde IBM, l'association entre les caractères ISO 8859/1 et les codes internes est définie par:

- le "codepage" 500/1 pour ce qui concerne les codes EBCDIC dans un environnement international; on en trouvera la définition complète à la fin de cet article;
- le "codepage" 850 pour les codes ASCII sur IBM PS/2 (le "codepage" généralement d'application sur les IBM-PC est identifié comme "codepage" 437);
- une série de "Country Extended Codepage", extensions des tables de codes EBCDIC précédemment employées dans divers pays, en général très proches de la table 500/1 pour les caractères accentués. Ces "codepage", dont le plus important est le 037/1 (USA/Canada) ne nous concernent pas aujourd'hui.

Les besoins d'affichage et d'impression de textes accentués ont amené le SEGI, au cours de l'installation de nouveaux équipements terminaux, à adopter progressivement la table 500/1 comme norme pour la codification des caractères accentués. C'est ainsi que tous les terminaux de marque IBM actuellement installés à l'Université sont alignés sur ce "codepage" 500/1 et sont par conséquent dotés des dispositifs nécessaires à l'affichage, à la frappe et à l'impression de l'ensemble des caractères utilisés dans la langue française. Il subsiste cependant aujourd'hui des caractères très particuliers figurant dans la table 500/1 qui sont absents des terminaux.

L'adoption de la table 500/1 a cependant introduit deux incompatibilités.

La première concerne les fichiers, précédemment ou encore aujourd'hui, constitués au départ de terminaux obéissant à la norme "belge" connue sous le nom de table Va. Il s'agit de terminaux de type 3270 qui disposent de quelques caractères accentués tels que le é, è, ù, à, mais pas de l'ensemble complet incluant les accents circonflexes, tréma, ... Pendant longtemps, ces terminaux ont constitué le seul moyen d'obtenir un texte accentué presque correct. A l'Université, les terminaux de ce type les plus répandus sont de marque ITT.

Caractères accentués dans la table Va		
Graphisme	Code Va	Caractère affiché pour ce code en table 500/1
é	'C0'	{
è	'D0'	}
ç	'E0'	\
ù	'6A'	9
à	'7C'	@

La seconde incompatibilité concerne quelques caractères très particuliers ([, ], !, ^, -, |, /) dont les codes internes en table 500/1 (et d'ailleurs également en table Va) diffèrent des codes "standard" EBCDIC.

Ces codes "standard" se présentent dans des fichiers composés sur des terminaux "standard" tels que les terminaux ITT à clavier QUERTY ou dotés de l'APL. Ils se rencontrent également dans la plupart des fichiers qui proviennent (moyennant conversion de code) du monde ASCII (terminaux ou micro-ordinateurs connectés sur IBM 7171, fichiers provenant de bandes ASCII). Il en est de même pour les données venant, via les réseaux internationaux tels que EARN/BITNET, d'installations américaines ou même européennes n'ayant pas encore adopté la table 500/1. Ils apparaissent également dans les produits qui utilisent depuis toujours de tels caractères : le remplacement de la barre de concaténation (!) par le point d'exclamation, ou du signe de négation (-) par l'accent circonflexe (^) est bien connu des programmeurs PL/I ou REXX.

Différences entre "codepage" international 500/1 et EBCDIC standard		
Graphisme	Code table 500/1	EBCDIC standard
[	'4A'	'AD' <sup>32</sup>
]	'5A'	'BD' <sup>32</sup>
!	'4F'	'5A'
^	'5F'	-
^	'BA'	'5F'
	'BB'	'4F'
[	'B0'	'4A'

<sup>32</sup> Les caractères [ et ] n'étaient pas disponibles en EBCDIC à l'origine; les codes indiqués sont ceux généralement employés et constituent une norme de fait

Les terminaux n'utilisant pas la table 500/1 se font de plus en plus rares à l'Université, les tables nationales anciennes (telles que la Va "belge") disparaissant petit à petit. La première incompatibilité se résorbe donc progressivement. La seconde incompatibilité par contre s'accroît par suite de deux phénomènes :

- les échanges entre les mondes ASCII et EBCDIC deviennent de plus en plus nombreux, notamment à cause de la diffusion des micro-ordinateurs
- les échanges de données et de courrier informatique avec les pays étrangers, et en particulier les USA, se développent de plus en plus grâce aux réseaux tels que EARN.

Il semble illusoire d'espérer rapidement une uniformisation complète des codes; il apparaît donc, et les évolutions récentes des produits IBM le confirment, que deux normes continueront à coexister.

### PRATIQUEMENT AU SEGI AUJOURD'HUI

- L'environnement homogène table 500/1 rassemble la quasi-totalité des terminaux 3270 et imprimantes de grappe de marque IBM. Tout fichier peut être entré, affiché, imprimé sur ces équipements de manière cohérente et peut contenir tous les caractères accentués de la langue française (et d'une manière générale tous les caractères accentués des langues occidentales, sous réserve de pouvoir les introduire au clavier). Les imprimantes Laser du SEGI utilisent également le "codepage" 500/1, soit par défaut (nouvelle imprimante Laser SIEMENS), soit moyennant une option à indiquer (imprimante Laser IBM 3812).
- Les terminaux de marque ITT utilisent la table Va *ou* les codes EBCDIC "standard". Le premier cas se présente en général sur les terminaux possédant un clavier AZERTY, le second sur les terminaux possédant un clavier QWERTY et disposant éventuellement du jeu de caractères APL.

- Un fichier constitué en table Va pourra être exploité dans l'environnement table 500/1 sans difficulté si aucun caractère accentué n'est utilisé mais demandera une conversion si le fichier contient des caractères accentués.
- Un fichier constitué en EBCDIC "standard", quelle que soit son origine, pourra être utilisé dans l'environnement table 500/1 sans conversion si les caractères [ , ], ! , ^ , ¬ , | , / n'y apparaissent pas, ou si l'affichage/impression d'un caractère "équivalent" ne gêne pas. Dans le cas contraire, le fichier devra faire l'objet d'une conversion.
- Le programme FTPSEGI permet l'échange de données entre un micro-ordinateur et les ordinateurs du SEGI aussi bien en table 500/1 qu'en EBCDIC "standard" (étendu à tous les caractères accentués utilisés en français).
- Pour l'impression au SEGI, la commande PRT3812 imprimera sur la 3812 en table 500/1 si l'option FONT C0BE10 ou FONT C0BE12 est indiquée, en EBCDIC standard (sans accent) sinon. L'imprimante Laser SIEMENS imprimera également correctement un texte en table 500/1. L'imprimante à impact utilise une chaîne dont le "character set" est limité aux symboles EBCDIC "standard" et ne dispose pas des minuscules. Les programmes de "publishing" utilisés sur les ordinateurs du SEGI tiennent bien entendu compte des caractéristiques des diverses imprimantes. Nous renvoyons à la description de ces produits pour les aspects particuliers qu'ils manifestent.
- Outils de conversion de code sur terminal :
  - En ce qui concerne l'affichage, la commande CMS SET OUTPUT xx xx permet de convertir le code envoyé sur la ligne de manière à provoquer l'affichage d'un caractère différent de celui normalement affiché.
  - Pour le clavier, l'opération inverse peut être obtenue en entrant la commande SET INPUT xx xx.

CODEPAGE 500, VERSION 1												
	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		&	-	ø	Ø	°	μ	[	{	}	\	0
-1		é	/	É	a	j	·	£	A	J	÷	1
-2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
-3	ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
-4	à	è	À	È	d	m	u	+	D	M	U	4
-5	á	í	Á	Í	e	n	v	§	E	N	V	5
-6	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
-7	â	ï	Å	Ï	g	p	x	¼	G	P	X	7
-8	ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
-9	ñ	ß	Ñ	à	i	r	z	¾	I	R	Z	9
-A	[	]	9	:	«	ª	i	¬	-	1	2	3
-B	.	\$	,	#	»	º	¿		ô	û	Ô	Û
-C	<	*	%	@	>	æ	—	-	ö	ü	Ö	Ü
-D	(	)	_	'	ÿ	,	ÿ	¨	ò	ù	Ò	Ù
-E	+	;	>	=	Õ	Æ	Ó	'	ó	ú	Ó	Ú
-F	!	^	?	"	#	E	;	z	õ	ÿ	Õ	

### Définitions des caractères du codepage 500/1

<b>40</b>	interword space	<b>50</b>	ampersand	<b>60</b>	hyphen
<b>41</b>	required space	<b>51</b>	e acute small	<b>61</b>	slash
<b>42</b>	a circumflex small	<b>52</b>	e circumflex small	<b>62</b>	A circumflex capital
<b>43</b>	a diaeresis/umlaut small	<b>53</b>	e diaeresis/umlaut small	<b>63</b>	A diaeresis/umlaut capital
<b>44</b>	a grave small	<b>54</b>	e grave small	<b>64</b>	A grave capital
<b>45</b>	a acute small	<b>55</b>	i acute small	<b>65</b>	A acute capital
<b>46</b>	a tilde small	<b>56</b>	i circumflex small	<b>66</b>	A tilde capital
<b>47</b>	a overcircle small	<b>57</b>	i diaeresis/umlaut small	<b>67</b>	A overcircle capital
<b>48</b>	c cedilla small	<b>58</b>	i grave small	<b>68</b>	C cedilla capital
<b>49</b>	n tilde small	<b>59</b>	s sharp small	<b>69</b>	N tilde capital
<b>4A</b>	left bracket	<b>5A</b>	right bracket	<b>6A</b>	vertical broken bar
<b>4B</b>	period, full stop	<b>5B</b>	dollar	<b>6B</b>	comma
<b>4C</b>	less-than	<b>5C</b>	asterisk	<b>6C</b>	percent
<b>4D</b>	left parenthesis	<b>5D</b>	right parenthesis	<b>6D</b>	continuous underscore, underline
<b>4E</b>	plus	<b>5E</b>	semicolon	<b>6E</b>	greater-than
<b>4F</b>	exclamation point	<b>5F</b>	circumflex	<b>6F</b>	question mark

<b>70</b>	o slash small	<b>AA</b>	exclamation point, inverted	<b>C8</b>	H capital
<b>71</b>	E acute capital	<b>AB</b>	question mark, inverted	<b>C9</b>	I capital
<b>72</b>	E circumflex capital	<b>AC</b>	D stroke capital and Eth Icelandic capital	<b>DA</b>	numeric decimal one superscript
<b>73</b>	E diaeresis/umlaut capital	<b>AD</b>	Y acute capital	<b>DB</b>	u circumflex small
<b>74</b>	E grave capital	<b>AE</b>	Thorn Icelandic capital	<b>DC</b>	u diaeresis/umlaut small
<b>75</b>	I acute capital	<b>AF</b>	registered trademark symbol	<b>DD</b>	u grave small
<b>76</b>	I circumflex capital	<b>A0</b>	micro symbol	<b>DE</b>	u acute small
<b>77</b>	I diaeresis/umlaut capital	<b>A1</b>	tilde	<b>DF</b>	y diaeresis/umlaut small
<b>78</b>	I grave capital	<b>A2</b>	s small	<b>D0</b>	right brace
<b>79</b>	grave	<b>A3</b>	t small	<b>D1</b>	J capital
<b>7A</b>	colon	<b>A4</b>	u small	<b>D2</b>	K capital
<b>7B</b>	number	<b>A5</b>	v small	<b>D3</b>	L capital
<b>7C</b>	at	<b>A6</b>	w small	<b>D4</b>	M capital
<b>7D</b>	apostrophe	<b>A7</b>	x small	<b>D5</b>	N capital
<b>7E</b>	equals	<b>A8</b>	y small	<b>D6</b>	O capital
<b>7F</b>	quotation marks	<b>A9</b>	z small	<b>D7</b>	P capital
<b>80</b>	O slash capital	<b>BA</b>	logical not, end of line	<b>D8</b>	Q capital
<b>81</b>	a small	<b>BB</b>	vertical bar, logical or	<b>D9</b>	R capital
<b>82</b>	b small	<b>BC</b>	macron	<b>EA</b>	numeric decimal two superscript
<b>83</b>	c small	<b>BD</b>	Diaeresis or umlaut	<b>EB</b>	O circumflex capital
<b>84</b>	d small	<b>BE</b>	acute	<b>EC</b>	O diaeresis/umlaut capital
<b>85</b>	e small	<b>BF</b>	multiply	<b>ED</b>	O grave capital
<b>86</b>	f small	<b>B0</b>	cent	<b>EE</b>	O acute capital
<b>87</b>	g small	<b>B1</b>	pound	<b>EF</b>	O tilde capital
<b>88</b>	h small	<b>B2</b>	yen	<b>E0</b>	reverse slash
<b>89</b>	i small	<b>B3</b>	middle dot	<b>E1</b>	divide
<b>8A</b>	left angle quotes	<b>B4</b>	copyright symbol	<b>E2</b>	S capital
<b>8B</b>	right angle quotes	<b>B5</b>	section symbol(USA), paragraph symbol(Europe)	<b>E3</b>	T capital
<b>8C</b>	eth Icelandic small	<b>B6</b>	paragraph symbol (USA)	<b>E4</b>	U capital
<b>8D</b>	y acute small	<b>B7</b>	numeric fraction one-quarter	<b>E5</b>	V capital
<b>8E</b>	thorn Icelandic small	<b>B8</b>	numeric fraction one-half	<b>E6</b>	W capital
<b>8F</b>	plus-or-minus	<b>B9</b>	numeric fraction three-quarters	<b>E7</b>	X capital
<b>90</b>	degree symbol	<b>CA</b>	underline superscript	<b>E8</b>	Y capital
<b>91</b>	j small	<b>CB</b>	o circumflex small	<b>E9</b>	Z capital
<b>92</b>	k small	<b>CC</b>	o diaeresis/umlaut small	<b>FA</b>	numeric decimal three superscript
<b>93</b>	l small	<b>CD</b>	o grave small	<b>FB</b>	U circumflex capital
<b>94</b>	m small	<b>CE</b>	o acute small	<b>FC</b>	U diaeresis/umlaut capital
<b>95</b>	n small	<b>CF</b>	o tilde small	<b>FD</b>	U grave capital
<b>96</b>	o small	<b>C0</b>	left brace	<b>FE</b>	U acute capital
<b>97</b>	p small	<b>C1</b>	A capital	<b>F0</b>	numeric decimal zero
<b>98</b>	q small	<b>C2</b>	B capital	<b>F1</b>	numeric decimal one
<b>99</b>	r small	<b>C3</b>	C capital	<b>F2</b>	numeric decimal two
<b>9A</b>	ordinal indicator - feminine	<b>C4</b>	D capital	<b>F3</b>	numeric decimal three
<b>9B</b>	ordinal indicator - masculine	<b>C5</b>	E capital	<b>F4</b>	numeric decimal four
<b>9C</b>	æ diphthong small	<b>C6</b>	F capital	<b>F5</b>	numeric decimal five
<b>9D</b>	cedilla	<b>C7</b>	G capital	<b>F6</b>	numeric decimal six
<b>9E</b>	Æ diphthong capital			<b>F7</b>	numeric decimal seven
<b>9F</b>	international currency				

**F8** numeric decimal eight

**F9** numeric decimal nine



/JOB 53  
/SET 54  
" 31  
\* 24  
&TRACE 38  
% 24  
= 33

## A

A 29  
abend 59  
ACCESS 17  
ACCOFF 110  
ACF2 73  
APL 52  
APL2 52  
Architecture 5  
ASCII 67, 121  
ASSEMBLE 18  
ASSEMBLER 52  
AUTOSAVE 35

## B

bande 67, 68  
BANDES 61  
bandes magnétiques 61  
bases de données relationnelles 98  
batch 18, 53, 59  
batch (limitations par classe) 85  
BATCH CANCEL 59  
BATCH HOLD 59  
BATCH NEWCLASS 59  
BATCH QUERY 58  
BATCH RELEASE 59  
BEGIN 8, 45  
bibliothèque cms 22  
BLIULG11 1  
BLKSIZE 22, 48  
BLOCK 22, 48  
BMDP 89, 93

BOTTOM 30  
break 45

## C

C 31, 51  
CALCOMP 89, 95, 110  
calcul symbolique 89, 107  
CAS 53  
CAS3490 53  
CASE MIXED 28  
cassettes 61  
CC 31  
CHANGE 11, 32  
Chargement d'un programme 46  
CLASS 53  
classe de travaux batch (limitations) 85  
Client/serveur 80  
CLOSE 10  
CMS 5, 13  
CMS Pipelines 43  
COBOL 51  
codepage 121  
COMMON dynamiques 49  
COMMUNICATION 77  
commuté 7  
compilateur 48  
compilation 45  
composants du réseau RSCS 78  
compress 22, 23  
comptes 3, 109  
Concentrateur 82  
connexion 9  
console 5, 8  
console stack 38, 41  
CONT 10  
contrat-disque 72, 109  
contrôle 3  
contrôle RSCS 79  
COPY 11  
COPYFILE 20  
COST 110  
COSTDISK 109, 110

COURRIER ELECTRONIQUE 80  
CP console FUNCTION mode 8  
CP mode 45  
CPREAD 45  
CREDIT 109, 110  
crédits 3, 109

## D

D 30  
DATA 18  
DCF 89, 98  
DCS 78, 111  
ddname 21  
DEFAULTS 21  
DEFINE 8  
DELSAVE 69  
Démarrage d'une machine virtuelle 73  
DETACH 9  
DIRECTORY 8, 70  
Dirmaint 70, 109  
DIS5080 96, 97, 98  
DISCALC 96, 97  
DISCARD 21  
DISCONNECT 7  
DISCONNECT automatique 8  
DISLOAD 96, 97  
DISPCTEK 96, 97  
disque 5  
DISSPLA 95, 96  
DMSGPI 52  
DMSOM 52  
documentation 3  
DOWN 30  
DROPBUF 38

## E

EARN 80  
EBCDIC 121  
échange de messages 78  
éclatement 28  
EDIT MODE 26  
éditeur 25  
édition 25

EMSG 14  
ERASE 19  
ESSL 49, 93  
EXEC 37, 38  
EXEC2 37  
EXECIO 38  
EXECUTE 25  
Exécution d'un programme 48  
EXTRACT 38

## F

F 31  
FFILE 35  
fichier 80  
fichier de noms 36  
FILE 29, 35  
FILEDEF 21, 22, 48, 64  
FILELIST 24  
filemode 17, 18  
filemode number 18, 22  
filetype 18  
FORM 11  
FORMAT 17, 41  
FORTRAN 18, 49  
FORTVS2 49  
frontaux 81  
FTP 81  
FTPSEGI 97  
fullscreen 90

## G

GAM 98  
GDDM 89, 95, 98  
gestion 3  
GET 33  
GETFMADR 38  
GETNDATA 58  
GLIM 89, 94  
GLOBAL MACLIB 45  
GLOBAL TXTLIB 46  
GLOBALV 38  
GLPOP 53  
GLPUSH 53

graphiques 95, 98

## H

HELP 35, 43  
HELP REXX MENU 39  
Help-Desk 3  
hold 7, 10, 53  
HOLDING 45  
HOUR 53  
HT 45  
HX 45

## I

IBM 96  
IBM 3172 82  
IBM 3270 89  
IBM 3745 82  
IBM 7171 82  
images 98  
impression 54  
incompatibilité 48  
INDICATE 13  
INPUT 27  
INPUT MODE 26  
INPUT ZONE 27  
INPUT-MODE 27  
insertion 28  
INTERNET 80  
interpréteur 48  
interpréteur de commandes 37  
INTERPRETEURS 37  
interruption 45  
IPL 13  
ISO 8859/1 121  
ISPF/PDF 89, 102

## J

JEUX DE CARACTERES 121  
JOB 53  
JOBHEADE 57  
JOBTRAIL 57  
jointure 28

## K

KERMIT 97

## L

LABELDEF 63  
langages de programmation 48  
LASER 60, 86, 111  
LINK 9, 71  
LISP 107, 108  
LISTFILE 19  
LISTING 18, 46  
LISTSAVE 68  
LNAME 37, 80, 81  
LOAD 46  
LOADLIB 18, 22, 23  
LOCATE 32  
Logiciels d'applications 89  
logiciels publics 9  
LOGOFF 7  
LOGOFF automatique 8  
LOGON 7  
LRECL 22, 48

## M

M 31  
MACLIB 18, 22, 45  
MACLIST 22, 23  
MACROS 52  
MAILBOOK 80, 81  
majuscules 28  
majuscules/minuscules 10

MAKEBUF 38  
memoire 5  
Message 14, 80  
message d'erreur 14  
micro-informatique 3  
minidisque 17, 70  
minidisque temporaire 8, 17  
minidisques 109, 110  
minuscules 28  
mire 7  
MM 31  
Modes de chargement 49  
Modification du mot de passe 73  
Modification du mot de passe associé à un "userid" 73  
MODULE 18  
MORE 45  
Mots de passe 73  
Mots de passe des machines dépendantes 73  
MOUNT 62  
MOVEFILE 22  
MSG 14, 80

## N

NAG 49, 89  
NAMES FILE 36  
Netscape 80  
NEWHour 59  
NEWSucces 59  
NFS 81  
NOCONT 10  
NOHOLD 10  
NOTE 37, 80  
NOTEBOOK 37  
numéro de message 14

## O

OS simulated data set 48  
OSMACRO 52  
OSPSI 52  
OSVSAM 52

## P

P 31  
PA1 45  
Parallélisation 50  
PARSE 41  
partitione 22  
PASCAL 50  
password 9  
PBX 82  
PCH 13  
PEEK 20  
PF key 14  
PFK 14, 28  
Pipelines 43  
PL/1 51  
PLIOPT 18, 51  
POST 80, 81  
POWER INPUT 100  
PRINT 20, 54  
printer 9, 10  
PROFILE 42  
program stack 38, 43  
Protection des fichiers 74  
Protection des minidisques 74  
Protection par terminal physique 74  
PRT 13  
PRT3812 60  
PUBLIB 89, 102  
PUBLISHI 98  
publishing 98  
PULL 41  
punch 8, 10, 20  
PURGE 11  
PUSH 41  
PW 53

## Q

QMF 98  
QQ 34  
QUIT 33  
QUERY 10, 14  
QUERY ALL 8  
QUERY DISK 17  
QUERY PK 28

QUEUE 41  
QUIT 33

## R

RAPPEL DES COMMANDES 15  
RB 55  
RDR 13  
RDRLIST 21  
reader 8, 10, 11  
RECBATCH 55  
RECEIVE 80  
RECFM 22  
recherche documentaire 89  
RECOVER 31  
RECTRAIL 57  
REDUCE 89, 107  
RELEASE 17  
RELTAPE 62  
RENAME 20  
répéter 33  
réseau 77  
réseau commuté 86  
RESEAU INTERACTIF 77  
réseau networking 79  
RESEAU RSCS 78  
réseaux 81  
RESERVE 18  
RESET 29, 31  
RESTORE 68  
restrictions 48  
RETRIEVE 15  
REXX 39  
RSCS 12, 78  
RT 45  
RUNNING 45

## S

S.A.S. 89  
SADT 38  
SAS 89, 95  
Sauvetage 67  
SAVE 35, 67  
SAY 41

SB 55  
scanner 98  
SCRIPT 89, 98  
Search Manager 89, 98  
Secrétariat 3  
Sécurité du système 73  
SENBATCH 55  
SENDFILE 21, 37, 80  
SET ABBREV 38  
SET AUTOSAVE 35  
SET CASE 28  
SET CMSTYPE 38  
SET EMSG 14  
SET EXECTRAC 38  
SET MSG 36  
SET NUMBER OFF 34  
SET NUMBER ON 34  
SET PFnn 14  
SET RETRIEVE 15  
SET RUN ON 7  
SET TABS 28  
SF 37  
SIGNAL ON ERROR 42  
SIGNAL ON HALT 42  
simulated data set 48  
SLEEP 8  
SORT 22, 34  
SOS LINEADD 30  
SPACE 71  
SPACE LIMIT 109  
SPACE MANAGER 67  
SPI 37  
SPOOL 5, 9, 10  
SPOOL limites 9  
spoolid 10  
SPPOP 53  
SPPUSH 53  
SPRECUP 69  
SQL 89, 98  
STATPACK 89, 95  
SUCCES 53

## T

T3390 8  
tableur 90  
tabulation 28  
TAG 12  
TAPE 64  
TAPE DUMP 65  
TAPE LOAD 65  
TAPE REW 65  
TAPE SCAN 66  
TAPE WTM 65  
TAPES 53, 61  
TCP/IP 77, 81  
TDISK 53  
TDSKCMS 17  
TDSKOS 17, 18  
TEKTRONIX 89  
télétype 45  
TELL 36, 80  
TELNET 81  
TERMINAL 14, 45  
terminal input buffer 38  
TEXT 18, 46  
TIME 54  
TOP 30  
touches fonctions 14  
TRACE 42  
traitement de texte 98  
TRAN5OFF 52  
TRAN5TOE 52  
TRANSFER 12  
transfert de fichiers 78  
TSP 89, 94  
TTIME 13  
TTY 45  
TVECTIME 13  
TXTLIB 18, 22, 23  
TYPE 19

## U

unit record 9  
Unix 80  
UP 30

## V

VAST-2 50  
VBS 22  
Vectorisation 49  
VM/ESA 5  
VM/XA 46  
VMFCLEAR 38  
VMREAD 45  
VS 22  
VSAM 17, 69  
VSAMLOAD 70  
VSAMSAVE 69  
VTAM 7  
VTAM/SNA 77  
VTIME 13  
VVECTIME 13

## W

WANTDISK 60  
WANTDUMP 60  
WEB 80

## X

X-WINDOWS 81  
X25 78  
XA 13, 46, 47, 49, 50  
XEDIT 25, 26  
XEDIT DISPLAY MODE 26