

SOLAR

RTESD

Moniteur temps réel disque

LOGICIEL

LOGICIEL

LOGICIEL

LOGICIEL

LOGICIEL

RTES-D

MANUEL D'UTILISATION

△ en haut de page indique le changement complet de la page
par rapport à l'IE précédent

| en marge indique la partie modifiée par rapport à l'IE pré-
cédent

AVANT PROPOS

Le lecteur trouvera essentiellement dans ce manuel des exemples pratiques de génération de RTES-D et de mise en oeuvre de son application.

Les chapitres qui le constituent font fréquemment référence à des notions de système ou de langage supposées acquises et qui ne sont que partiellement reprises.

Si ce manuel est donc en quelque sorte le dernier à lire, son contenu n'en est pas moins de première importance.

SOMMAIRE	Page
1 - INTRODUCTION	1-1
1.1 - RTES-D système de 64K mots à 1024K mots	1-1
1.2 - RTES-D et la périphérie industrielle	1-1
1.3 - RTES-D et les langages évolués	1-2
1.4 - RTES-D et l'application	1-2
2 - PRODUCTION EN "BACKGROUND" D'UNE TACHE DE L'APPLICATION	2-1
2.1 - Caractéristiques de cette tâche	2-1
2.2 - Quelques points relatifs à la programmation	2-2
3 - BIBLIOTHEQUE TEMPS REEL	3-1
3.1 - Introduction	3-1
3.2 - Syntaxe	3-1
3.2.1 - Syntaxe FORTRAN	3-1
3.2.2 - Syntaxe PL 16	3-1
3.3 - Passage des paramètres	3-2
3.4 - Fonctions réalisées	3-2
3.5 - Pré-détection d'erreurs dans un sous-programme	3-2
3.5.1 - Requêtes sur liste d'événements	3-2
3.5.2 - Liste des paramètres incomplète	3-3
3.6 - Comment exploiter le compte rendu de requête	3-3
3.6.1 - Requêtes s'étant déroulée en tout ou en partie	3-3
3.6.2 - Erreurs détectées par le sous-programme	3-3
3.7 - Composition de la bibliothèque	3-3
3.8 - Mots réservés du langage PL16	3-4
3.9 - Utilisation de la bibliothèque	3-4
3.9.1 - Mise en oeuvre	3-4
3.9.2 - Editions de liens d'un programme	3-4
3.10 - Fonctionnement de la bibliothèque	3-5
4 - UTILISATION DES DISQUES SOUS LES SYSTEMES DISQUE STANDARD	4-1
4.1 - Rappels	4-1
4.1.1 - Caractéristiques techniques des disques à têtes fixes Modules FHD01 A-0	4-1
4.1.2 - Caractéristiques techniques des disques à tête mobile Modules CDB10-0	4-1

	Page
4.1.3 - Notion de volume, d'espace et d'unité fonctionnelle disque	4-1
4.2 - Utilisation de IOCS	4-2
4.3 - Utilisation de FMS	4-2
4.4 - Exploitation sous les systèmes disques standards	4-2
4.5 - Principe d'utilisation des disques à tête mobile	4-3
4.5.1 - Initialisation du disque	4-3
4.5.2 - Opération de montage de volume	4-3
4.6 - Gestion des cylindres en défaut	4-4
4.7 - Configuration en FU des disques	4-4
5 - GENERATION D'UN SYSTEME RTES-D	5-1
5.1 - Principe	5-1
5.2 - GENIO configuration de IOCS	5-1
5.3 - GENFMS configuration de FMS	5-2
5.3.1 - Principe	5-2
5.3.2 - Configuration d'une FU gérée par FMS	5-3
5.3.3 - Exemple de macro-instructions de GENFMS	5-5
5.4 - GENCOM génération de RTES-D	5-5
5.4.1 - Macro-instructions spécifiques de RTES-D	5-7
5.4.2 - Remarques	5-7
5.4.3 - Exemple de génération de RTES-D	5-7
6 - UTILISATION DU MONITEUR "BACKGROUND"	6-1
6.1 - Les requêtes de BACKM	6-1
6.2 - Les commandes de BACKM	6-1
6.3 - Les processeurs de BACKM	6-3
6.4 - Mise en oeuvre de BACKM	6-7
6.5 - Les numéros des erreurs détectées par BACKM	6-8
7 - DIALOGUE INITIAL DE CONFIGURATION	7-1
8 - CONFIGURATION D'UNE APPLICATION	8-1
8.1 - Intégration de tâches et de processeurs	8-1
8.2 - Intégration et lancement du moniteur "background"	8-1
8.3 - Impressions d'état du système	8-1
8.4 - Sauvegarde d'applications sur disque	8-2
8.5 - Diagramme de temps	8-3
8.6 - Lancement de l'application	8-4
9 - TRAITEMENT DU DEFAUT SECTEUR ET DU "RESTART" AUTOMATIQUE	9-1
A - ANNEXE	A-1

1 - INTRODUCTION

RTES-D permet l'exploitation d'une installation ayant une taille mémoire comprise entre 64K mots et 1024K mots. Il utilise le système d'entrées-sorties IOCS et les modules d'échanges des périphériques conventionnels et industriels développés sur le SOLAR 16. Par l'intermédiaire du système de fichiers FMS et de ses diverses méthodes d'accès, il assure l'exploitation d'un (de plusieurs) disque(s) à têtes fixes et/ou disque(s) à cartouches (jusqu'à 4 unités par coupleur pour les disques à cartouches).

Ce manuel présente une "check-list" des opérations concernant la mise en oeuvre du système RTES-D, du moniteur d'enchaînement de travaux ou moniteur "Background" intégré en option sous RTES-D, et donne quelques exemples simples de réalisation de tâches de l'application en langage PL16 et FORTRAN temps réel.

1.1 - RTES-D, SYSTEME DE 64K MOTS A 1024K MOTS

Lors de la génération du système, l'espace-mémoire utilisateur est découpé en partitions dont le nombre (64 au maximum pour RTES-D) et la taille (64 K mots maximum) sont configurés aux besoins de l'application.

En règle générale, les tâches intégrées dans ces partitions sont en mode esclave : cette règle est impérative pour les tâches (résidentes en mémoire centrale ou non résidentes) susceptibles d'être exécutées au delà de 32 K.

Les adresses 32K-1 ('7FFF) ou 32K ('8000) ne constituent pas des frontières physiques : une partition peut être située "à cheval" sur ces adresses, par exemple commencer en '7800 et se terminer en '8200 ; une telle partition ne pourra contenir que des tâches en mode esclave. Par contre une partition ne peut être implantée à cheval sur 2 pages de 64K.

Le moniteur "background" BACKM s'exécutant en mode maître, il conviendra sur les applications de plus de 32K, de lui réserver une partition de 2K mots minimum en zone maître (entièrement incluses dans la zone [0, 32K - 1]).

Cette partition sera de type résident ou non résident en mémoire centrale.

1.2 - RTES-D ET LA PERIPHERIE INDUSTRIELLE

Les modules d'échanges (ou drivers) des périphériques industriels développés sur le SOLAR 16 peuvent être intégrés au système IOCS et utilisés sous RTES-D. On citera par exemple :

- le module DIT16 16 appels externes
- le module IML05 chaîne de mesures 50 voies/seconde
- le module DOL32 32 sorties tout ou rien
- le module DIL48 48 entrées tout ou rien

Le lecteur trouvera dans les "Manuels d'exploitation des Périphériques" la documentation nécessaire à la mise en oeuvre de ces modules.

1.3 - RTES-D ET LES LANGAGES EVOLUES

La bibliothèque BIBRTE-RT des sous-programmes d'interface langage système permet l'écriture de tâches à caractéristique temps réel en FORTRAN ou PL16. Cette bibliothèque donne accès aux services spécifiques de RTES-D.

- CALL START : appel de tâche avec délai initial et/ou période
- CALL WAIT : temporisation d'une tâche

D'autres sous programmes réalisent les interfaces avec les modules d'entrées-sorties industrielles ; la documentation sur l'utilisation de ces sous-programmes se trouve dans le manuel d'utilisation des "drivers" IOCS. Ce sont par exemple :

- CALL AIRD : entrée analogique aléatoire
- CALL AISQ : entrée analogique séquentielle
- CALL DI : entrée digitale.

1.4 - RTES-D ET L'APPLICATION

Ci-jointes quelques caractéristiques "logiciel" d'une application sous RTES-D.

- Nombre maximum de tâches utilisateur :
 - 120 tâches de type logiciel, résidentes ou non résidentes, avec ou sans structure "d'overlay" de priorité comprise entre 10 et 126 inclus, ou égale à 3, 4 et 5.
 - 14 tâches matériel (résidentes) de priorité comprise entre 1 et 15 et différente de celles gérées par IOCS.
- Nombre maximum de systèmes ou applications stockées sur disque et directement "bootstrappable" en mémoire centrale : 8.
- Le système 0 est (impérativement) BOS-D
- Le système 1 est (impérativement) RTES-D "vierge" (ou non configuré)
- Les systèmes 2 à 7 peuvent être :
 - . d'autres : RTES-D "vierge" (non configuré)
 - . mais surtout des applications différentes issues d'un ou plusieurs RTES-D vierges (des jeux différents des tâches de la ou des applications).

2 - PRODUCTION EN "BACKGROUND" D'UNE TACHE DE L'APPLICATION

A partir de l'exemple, simple, d'une tâche écrite en PL16, produite en "background" et intégrée sous RTES-D, on a mis l'accent dans les pages qui suivent, sur quelques points pratiques de programmation ou de mode opératoire intervenant dans la réalisation d'une tâche.

Ecrit dans un style très proche du langage machine, cet exemple de programme PL 16 peut également donner quelques indications sur la programmation d'une tâche en Assembleur ASM16.

2.1 - CARACTERISTIQUES DE CETTE TACHE

Elle a une structure d'"overlay" comprend :

- la racine (MAIN PROCEDURE TOVER)
- trois branches (SEGMENT PROCEDURE TOVER1, TOVER2 et TOVER3).

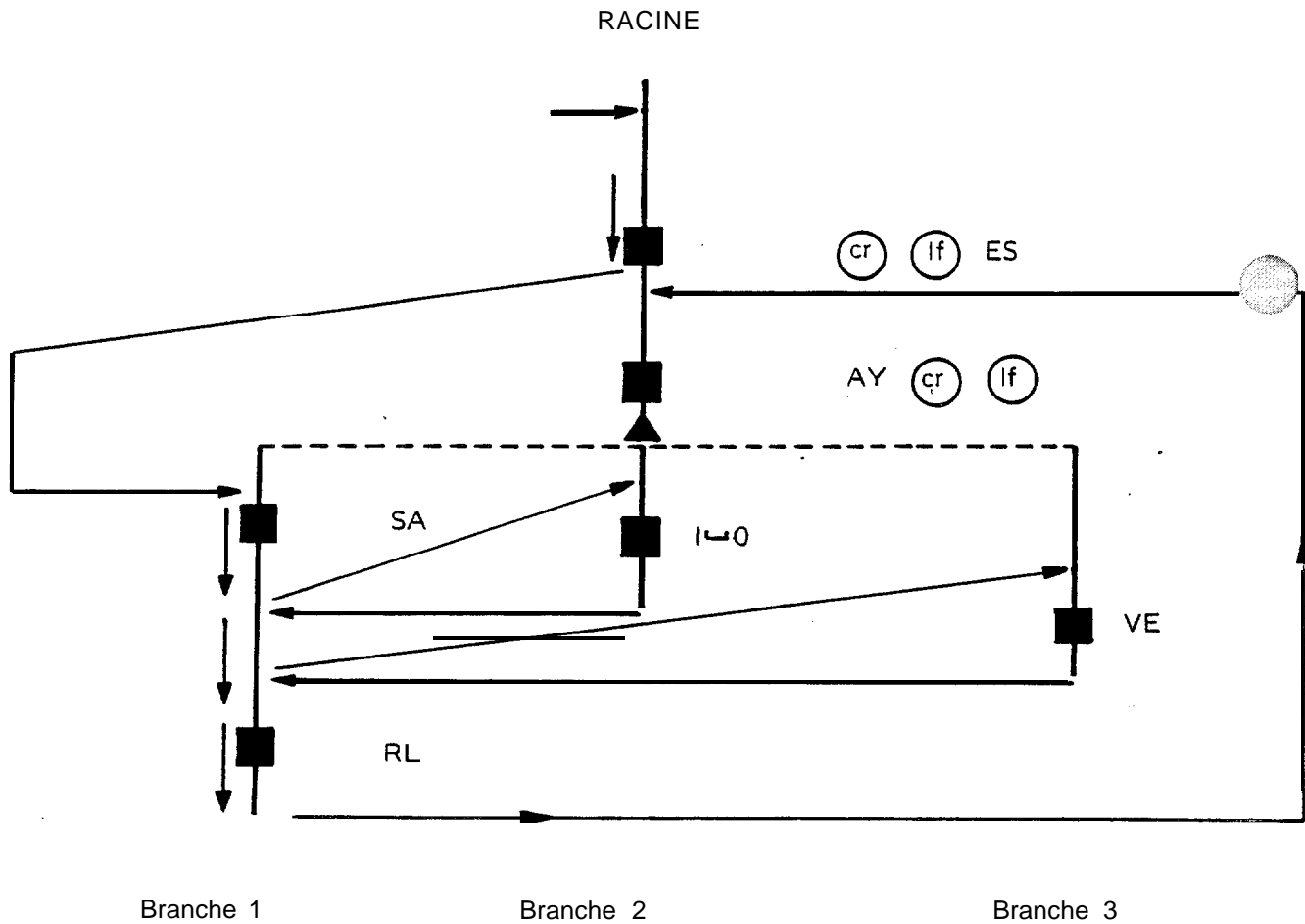
Son but global est d'imprimer le message :

 (cr) (1f) ESSAI "OVERLAY" (cr) (1f)

à chaque activation.

L'impression de ce message est fragmentée selon le schéma suivant :

- la racine (TOVER) imprime (cr) (1f) ES , puis appelle la branche 1 (TOVER1).
- cette branche imprime SA puis appelle la branche 2 (TOVER 2)
- celle-ci imprime une autre partie du message, LO puis retourne à la branche 1.
- la branche 3 (TOVER 3) est alors appelée : elle imprime VE puis retourne à la branche 1.
- cette branche imprime RL et retourne à la racine qui sort la fin du message.



2.2 - QUELQUES POINTS RELATIFS A LA PROGRAMMATION

Le lecteur suivra les messages et commentaires destinés à en faciliter la compréhension et à éclairer des points importants de réalisation.

MONITEUR BACKGROUND

#/JOB TCV,AA,D2,,12

MONITEUR BACKGROUND

```
#/MSG -C O M P I L A T I O N - D E - L A - R A C I N E --  
#/CALL PL  
#/SI CR  
#/LD LP  
#/IPLC
```

MONITEUR BACKGROUND

LIGNE	RL	RL	RW	DUNNEES	TABL	PROG	PF	BN	:*SOURCE
0001	1F	1F	1F	1F	0000	0000	00	00	: <<----->>
0002	1F	1F	1F	1F	0000	0000	00	00	: << PROCEDURE P15 >>
0003	1F	1F	1F	1F	0000	0000	00	00	: << >>
0004	1F	1F	1F	1F	0000	0000	00	00	: << RACINE >>
0005	1F	1F	1F	1F	0000	0000	00	00	: <<----->>
0006	1F	1F	1F	1F	0000	0000	00	00	: MAIN PROCEDURE TOVER
0007	1F	1F	1F	1F	0000	0000	00	00	: .KSTORE SECTION KTOV
0008	1F	1F	1F	02	0000	0000	01	01	: RES 90;<<-----1
0009	1F	1F	1F	02	0050	0000	00	01	: <<
0010	1F	1F	1F	02	0050	0000	01	01	: << DIMENSIONNER LA KSTORE EN FONCTION DES
0011	1F	1F	1F	02	0050	0000	01	01	: << BESOINS ET DES REQUETES UTILISEES
0012	1F	1F	1F	02	0050	0000	01	01	: .COMMON SECTION COMTOV
0013	03	1F	1F	03	0000	0000	01	01	: INSTRUCTION SVC (3,'IC00);
0014	03	1F	1F	03	0000	0000	01	01	: EXT PROCEDURE TOVER;<<-----2
0015	03	1F	1F	03	0000	0000	01	01	: << DECLARATION D'EXTERNE NECESSAIRE POUR
0016	03	1F	1F	03	0000	0000	01	01	: << INITIALISATION CORRECTE DE L'ADRESSE DE
0017	03	1F	1F	03	0000	0000	01	01	: << LANCEMENT DE LA TACHE DANS LA PST
0018	03	1F	1F	03	0000	0000	01	01	: << CF LIBELLE 3 CI-DESSOUS
0019	03	1F	1F	03	0000	0000	01	01	: <<
0020	03	1F	1F	03	0000	0000	01	01	: WORD FCB0B1=('TO'),
0021	03	1F	1F	03	0001	0000	01	01	: FCB1B1=('VE'),
0022	03	1F	1F	03	0002	0000	01	01	: FCB2B1=('R1');
0023	03	1F	1F	03	0003	0000	01	01	: CONSTANT IUCS=8,
0024	03	1F	1F	03	0003	0000	01	01	: CEXIT=23,
0025	03	1F	1F	03	0003	0000	01	01	: BCHLR=61,
0026	03	1F	1F	03	0003	0000	01	01	: OUT='2000,
0027	03	1F	1F	03	0003	0000	01	01	: EMOB='9000,
0028	03	1F	1F	03	0003	0000	01	01	: TTE='88;
0029	03	1F	1F	03	0003	0000	01	01	: ARRAY 2 WORD MESTOVE=('8DOA,'ES');
0030	03	1F	1F	03	0004	0002	00	01	: WORD IUCBTOVE0=(OUT+EMOB+TTE),
0031	03	1F	1F	03	0005	0002	00	01	: IUCBTOVE1=(0),
0032	03	1F	1F	03	0006	0002	00	01	: IUCBTOVE2=(4),
0033	03	1F	1F	03	0007	0002	00	01	: IUCBTOVE3=(0),
0034	03	1F	1F	03	0008	0002	00	01	: IUCBTOVE4=(0),
0035	03	1F	1F	03	0009	0002	00	01	: IUCBTOVE5=(0);
0036	03	1F	1F	03	000A	0002	00	01	: ARRAY 2 WORD FINTOVE=('AY','8DOA');
0037	03	1F	1F	03	000B	0004	00	01	: .LOCAL SECTION LOCTOV
0038	03	04	1F	04	0000	0004	00	01	: SOFT TASK 25 I
0039	03	04	1F	04	0000	0004	00	01	: RC=COMTOV,
0040	03	04	1F	04	0005	0004	00	01	: RL=LOCTOV,
0041	03	04	1F	04	0006	0004	00	01	: RK=KTOV,
0042	03	04	1F	04	0009	0004	00	01	: START=TOVER,<<-----3
0043	03	04	1F	04	0009	0004	00	01	: << INITIALISATION ADRESSE LANCEMENT DE LA TACHE
0044	03	04	1F	04	0009	0004	00	01	: <<
0045	03	04	1F	04	0009	0004	00	01	: XTNUSR=25*256, << NT=25 USR=0-----4
0046	03	04	1F	04	000D	0004	00	01	: << INITIALISATION DES 4 MOTS EXTENSION DE PST
0047	03	04	1F	04	000D	0004	00	01	: << LE PROGRAMME CONSTRUIT A PARTIR DE CES 4
0048	03	04	1F	04	000D	0004	00	01	: << PROCEDURES COMPORTE UNE SEULE DECLARATION
0049	03	04	1F	04	000D	0004	00	01	: << DE TACHE;C'EST UNE TACHE EN MODE ESCLAVE,
0050	03	04	1F	04	000D	0004	00	01	: << DE TYPE SOFTWARE ET DE PRIORITE 25
0051	03	04	1F	04	000D	0004	00	01	: << RAPPEL:
0052	03	04	1F	04	000D	0004	00	01	: << NT (0,119)
0053	03	04	1F	04	000D	0004	00	01	: << USR /=128,129
0054	03	04	1F	04	000D	0004	00	01	: << XAPPMAX (0,255)
0055	03	04	1F	04	000D	0004	00	01	: << XSYSTEM RESERVE

MONITEUR BACKGROUND

LIGNE	RC	RL	RW	DONNEES	TABL	PRG	PF	BN	°SOURCE
0056	03	04	1F	04	000D	0004	0000	01 01 :	XAPPMAX=10, << CUMUL MAXI D'APPELS = 10
0057	03	04	1F	04	000E	0004	0000	01 01 :	XPARTITION=15, << NUMERO DE PARTITION = 15
0058	03	04	1F	04	000F	0004	0000	01 01 :	XSYSTEM=0;
0059	03	04	1F	04	0010	0004	0000	01 01 :	.USING RC=COMTOV,RL=LOCTOV,RK=KTOV;<<-----5
0060	03	04	1F	04	0010	0004	0007	01 01 :	<< INITIALISATION DES BASES POUR LA PREMIERE
0061	03	04	1F	04	0010	0004	0007	01 01 :	<< ACTIVATION
0062	03	04	1F	04	0010	0004	0007	01 01 :	<< INITIALISATION INUTILE SI LES BASES SONT
0063	03	04	1F	04	0010	0004	0007	01 01 :	<< CHARGÉES DANS LA PST LORS DE LA DECLARATION
0064	03	04	1F	04	0010	0004	0007	01 01 :	<< DE LA TACHE
0065	03	04	1F	04	0010	0004	0007	01 01 :	<<
0066	03	04	1F	04	0010	0004	0007	01 01 :	AVANTI :
0067	03	04	1F	04	0010	0004	0007	01 01 :	RA:=@MESTOVE(0);
0068	03	04	1F	04	0010	0004	0009	01 01 :	IOCBOVE1:=RA;
0069	03	04	1F	04	0010	0004	000A	01 01 :	RA:=@IOCBOVE0; <<IMPRESSION DE (RC)+(LF)+ES
0070	03	04	1F	04	0010	0004	000B	01 01 :	SVC (IOC5);
0071	03	04	1F	04	0010	0004	000C	01 01 :	RA:=@FCBOB1; <<APPEL BRANCHE B1
0072	03	04	1F	04	0010	0004	000D	01 01 :	SVC (BCHLR);
0073	03	04	1F	04	0010	0004	000E	01 01 :	GOTO EROVER ON (RA>1);
0074	03	04	1F	04	0010	0004	0010	01 01 :	RA:=@FINTOVE(0); <<IMPRESSION DE AY+(RC)+(LF)
0075	03	04	1F	04	0010	0004	0012	01 01 :	IOCBOVE1:=RA;
0076	03	04	1F	04	0010	0004	0013	01 01 :	RA:=@IOCBOVE0;
0077	03	04	1F	04	0010	0004	0014	01 01 :	SVC (IOC5);
0078	03	04	1F	04	0010	0004	0015	01 01 :	EROVER :
0079	03	04	1F	04	0010	0004	0015	01 01 :	SVC (CEXIT);<<-----6
0080	03	04	1F	04	0010	0004	0016	01 01 :	<< FIN DE TACHE
0081	03	04	1F	04	0010	0004	0016	01 01 :	<< POUR UNE TACHE EN OVERLAY,CETTE REQUETE
0082	03	04	1F	04	0010	0004	0016	01 01 :	<< DOIT ETRE REALISEE DANS LA RACINE
0083	03	04	1F	04	0010	0004	0016	01 01 :	<< PROCHAINE INSTRUCTION EXECUTEE LORS D'UN
0084	03	04	1F	04	0010	0004	0016	01 01 :	<< APPEL ULTERIEUR:GOTO AVANTI
0085	03	04	1F	04	0010	0004	0016	01 01 :	<<
0086	03	04	1F	04	0010	0004	0016	01 01 :	GOTO AVANTI;
0087	03	04	1F	04	0010	0004	0017	01 01 :	END.

FIN DE COMPILATION 0000 ERREUR(S) 0097 MOTS

#/MSG - C O M P I L A T I O N - D E - L A - B R A N C H E - 1 --
#/CALL PL
#/SI CR
#/LD LP
#/IPLC

MONITEUR BACKGROUND

LIGNE	RC	RL	RH	CONNEES	TABL	PRDG	PF	BN	:=SOURCE
0001	1F	1F	1F	1F	0000	0000	00	00	: <<----->>
0002	1F	1F	1F	1F	0000	0000	00	00	: << P R O C E D U R E P 1 5 >>
0003	1F	1F	1F	1F	0000	0000	00	00	: << BRANCHE 1 >>
0004	1F	1F	1F	1F	0000	0000	00	00	: <<----->>
0005	1F	1F	1F	1F	0000	0000	00	00	: SEGMENT PROCEDURE TOVER1
0006	1F	1F	1F	1F	0000	0000	00	00	: .LOCAL SECTION LVERB1
0007	1F	02	1F	02	0000	0000	01	01	: INSTRUCTION SVC (3,'1C00);
0008	1F	02	1F	02	0000	0000	01	01	: CONSTANT IOCS=8,
0009	1F	02	1F	02	0000	0000	01	01	: BCHLR=61,
0010	1F	02	1F	02	0000	0000	01	01	: BACKR=63,
0011	1F	02	1F	02	0000	0000	01	01	: OUT='2000,
0012	1F	02	1F	02	0000	0000	01	01	: EMOD='8000,
0013	1F	02	1F	02	0000	0000	01	01	: TTE='88,
0014	1F	02	1F	02	0000	0000	01	01	: BACK=62;
0015	1F	02	1F	02	0000	0000	01	01	: EXT PROCEDURE TOVER1;
0016	1F	02	1F	02	0000	0000	01	01	: ARRAY 1 WORD ENAVAN=(@TOVER1);<<-----7
0017	1F	02	1F	02	0001	0001	01	01	: << LE PREMIER MOT D'UNE BRANCHE (SECTION TABLE)
0018	1F	02	1F	02	0001	0001	01	01	: << DOIT CONTENIR SON ADRESSE DE LANCEMENT
0019	1F	02	1F	02	0001	0001	01	01	: <<
0020	1F	02	1F	02	0001	0001	01	01	: WORD IOCB0B1=(OUT+EMOD+TTE),
0021	1F	02	1F	02	0002	0001	01	01	: IOCB1B1=(1),
0022	1F	02	1F	02	0003	0001	01	01	: IOCB2B1=(2),
0023	1F	02	1F	02	0004	0001	01	01	: IOCB3B1=(0),
0024	1F	02	1F	02	0005	0001	01	01	: IOCB4B1=(0),
0025	1F	02	1F	02	0006	0001	01	01	: IOCB5B1=(0);
0026	1F	02	1F	02	0007	0001	01	01	: WORD FCBOB2=("T0"),
0027	1F	02	1F	02	0008	0001	01	01	: FCB1B2=("VE"),
0028	1F	02	1F	02	0009	0001	01	01	: FCB2B2=("R2");
0029	1F	02	1F	02	000A	0001	01	01	: WORD FCBOB3=("T0"),
0030	1F	02	1F	02	000B	0001	01	01	: FCB1B3=("VE"),
0031	1F	02	1F	02	000C	0001	01	01	: FCB2B3=("R3");
0032	1F	02	1F	02	000D	0001	01	01	: ARRAY 1 WORD SA1=("SA");
0033	1F	02	1F	02	000E	0002	01	01	: WORD RRL1=("RL");
0034	1F	02	1F	02	000F	0002	01	01	: .USING RL=LVERB1;<<-----8
0035	1F	02	1F	02	000F	0002	01	01	: << SAUVEGARDE DE LA BASE RL DE L'APPELANT,
0036	1F	02	1F	02	000F	0002	01	01	: << RACINE DU BRANCHE,ET INITIALISATION DE RL
0037	1F	02	1F	02	000F	0002	01	01	: <<
0038	1F	02	1F	02	000F	0002	01	01	: RA:=@SA1(0);
0039	1F	02	1F	02	000F	0002	01	01	: IOCB1B1:=RA;
0040	1F	02	1F	02	000F	0002	01	01	: RA:=@IOCB0B1; <<IMPRESSION DE SA
0041	1F	02	1F	02	000F	0002	01	01	: SVC (IOCS);
0042	1F	02	1F	02	000F	0002	01	01	: RA:=@FCBOB2;
0043	1F	02	1F	02	000F	0002	01	01	: SVC (BCHLR); <<CHARGEMENT BRANCHE B2 ET LANCEMENT
0044	1F	02	1F	02	000F	0002	01	01	: GOTO SORB1 ON (RA>1);
0045	1F	02	1F	02	000F	0002	01	01	: RA:=@FCBOB3;
0046	1F	02	1F	02	000F	0002	01	01	: SVC (BCHLR); <<CHARGEMENT BRANCHE B3 ET LANCEMENT
0047	1F	02	1F	02	000F	0002	01	01	: GOTO SORB1 ON (RA>1);
0048	1F	02	1F	02	000F	0002	01	01	: IOCB1B1:=@RRL1;
0049	1F	02	1F	02	0010	0002	01	01	: RA:=@IOCB0B1;
0050	1F	02	1F	02	0010	0002	01	01	: SVC (IOCS);
0051	1F	02	1F	02	0010	0002	01	01	: SORB1:
0052	1F	02	1F	02	0010	0002	01	01	: RESTORE (RL);<<-----9
0053	1F	02	1F	02	0010	0002	01	01	: << RESTAURATION DE LA BASE RL DE L'APPELANT
0054	1F	02	1F	02	0010	0002	01	01	: <<
0055	1F	02	1F	02	0010	0002	01	01	: SVC (BACKR); <<RETOUR RACINE

```
LIGNE RC RL RW DONNEES TABL PROG PF BN :*SOURCE
0056 1F 02 1F 02 0010 0002 001A 01 01 : ERB1:
0057 1F 02 1F 02 0010 0002 001A 01 01 : GOTO ERB1:
0058 1F 02 1F 02 0010 0002 001B 01 01 : END.
FIN DE COMPILATION 0000 ERREUR(S) 002F MOTS
```



```
#MSG -C O M P I L A T I O N - D E - L A - B R A N C H E - 2 --  
#/CALL PL  
#/SI CR  
#/LO LP  
#/IPLC
```

MONITEUR BACKGROUND

LIGNE	RC	RL	RW	DONNEES	TABL	PRDG	PF	BN	*SOURCE
0001	1F	1F	1F	1F	0000	0000	00	00	<<----->>
0002	1F	1F	1F	1F	0000	0000	00	00	<< P R O C E D U R E P 1 5 >>
0003	1F	1F	1F	1F	0000	0000	00	00	<< BRANCHE 2 >>
0004	1F	1F	1F	1F	0000	0000	00	00	<<----->>
0005	1F	1F	1F	1F	0000	0000	00	00	SEGMENT PROCEDURE TOVER2
0006	1F	1F	1F	1F	0000	0000	00	00	.LOCAL SECTION LVERB2
0007	1F	02	1F	02	0000	0000	00	01	INSTRUCTION SVC (3,'1C00);
0008	1F	02	1F	02	0000	0000	00	01	CONSTANT IDC5=8,
0009	1F	02	1F	02	0000	0000	00	01	BCHLR=61,
0010	1F	02	1F	02	0000	0000	00	01	OUT='2000,
0011	1F	02	1F	02	0000	0000	00	01	EMDD='9000,
0012	1F	02	1F	02	0000	0000	00	01	TTE='88,
0013	1F	02	1F	02	0000	0000	00	01	BACK=62;
0014	1F	02	1F	02	0000	0000	00	01	EXT PROCEDURE TOVER2;
0015	1F	02	1F	02	0000	0000	00	01	ARRAY 1 WORD ENAVAN=(@TOVER2);
0016	1F	02	1F	02	0001	0001	00	01	ARRAY 2 WORD MESIO=('I ',' N');
0017	1F	02	1F	02	0002	0003	00	01	WORD IOCB0B2=(OUT+EMDD+TTE),
0018	1F	02	1F	02	0003	0003	00	01	IOCB1B2=(@MESIO AND '7FFF),
0019	1F	02	1F	02	0004	0003	00	01	IOCB2B2=(4),
0020	1F	02	1F	02	0005	0003	00	01	IOCB3B2=(0),
0021	1F	02	1F	02	0006	0003	00	01	IOCB4B2=(0),
0022	1F	02	1F	02	0007	0003	00	01	IOCB5B2=(0);
0023	1F	02	1F	02	0008	0003	00	01	.USING RL=LVERB2;
0024	1F	02	1F	02	0008	0003	00	01	RA:=@IOCB0B2;
0025	1F	02	1F	02	0008	0003	00	01	SVC (IDC5);
0026	1F	02	1F	02	0009	0003	00	01	RESTORE (RL);
0027	1F	02	1F	02	0008	0003	00	01	SVC (BACK);
0028	1F	02	1F	02	0008	0003	00	01	ERB2:
0029	1F	02	1F	02	0008	0003	00	01	GOTO ERB2;
0030	1F	02	1F	02	0008	0003	00	01	END.

<<IMPRESSION DE I O

FIN DE COMPILATION 0000 ERREUR(S) '0019 MOTS

```
#MSG -C O M P I L A T I O N - D E - L A - B R A N C H E - 3 --  
#/CALL PL  
#/SI CR  
#/LO LP  
#/IPLC
```

MONITEUR BACKGROUND

LIGNE	RC	RL	RW	CUNNEES	TABL	PRUG	PF	BN	:=SOURCE
0001	1F	1F	1F	1F	0000	0000	00	00	: <<----->>
0002	1F	1F	1F	1F	0000	0000	00	00	: << P R O C E D U R E P 1 5 >>
0003	1F	1F	1F	1F	0000	0000	00	00	: << BRANCHE 3 >>
0004	1F	1F	1F	1F	0000	0000	00	00	: <<----->>
0005	1F	1F	1F	1F	0000	0000	00	00	: SEGMENT PROCEDURE TOVER3
0006	1F	1F	1F	1F	0000	0000	00	00	: .LOCAL SECTION LVERB3
0007	1F	02	1F	02	0000	0000	01	01	: INSTRUCTION SVC (3,'100);
0008	1F	02	1F	02	0000	0000	01	01	: CONSTANT IUCS=8,
0009	1F	02	1F	02	0000	0000	01	01	: BCHLR=61,
0010	1F	02	1F	02	0000	0000	01	01	: OUT='2000,
0011	1F	02	1F	02	0000	0000	01	01	: EMOO='R000,
0012	1F	02	1F	02	0000	0000	01	01	: TTE='88,
0013	1F	02	1F	02	0000	0000	01	01	: BACK=62;
0014	1F	02	1F	02	0000	0000	01	01	: EXT PROCEDURE TOVER3;
0015	1F	02	1F	02	0000	0000	01	01	: ARRAY 1 WORD ENAVAN=@TOVER3);
0016	1F	02	1F	02	0001	0001	00	01	: WORD VE=('VE');
0017	1F	02	1F	02	0002	0001	00	01	: WORD I0CB0B3=(OUT+EMOO+TTE),
0018	1F	02	1F	02	0003	0001	00	01	: I0CB1B3=@VE,
0019	1F	02	1F	02	0004	0001	00	01	: I0CB2B3={2},
0020	1F	02	1F	02	0005	0001	00	01	: I0CB3B3={0},
0021	1F	02	1F	02	0006	0001	00	01	: I0CB4B3={0},
0022	1F	02	1F	02	0007	0001	00	01	: I0CB5B3={0};
0023	1F	02	1F	02	0008	0001	00	01	: .USING RL=LVERB3;
0024	1F	02	1F	02	0008	0001	00	01	: RA=@I0CB0B3; <<IMPRESSION DE VE
0025	1F	02	1F	02	0008	0001	00	01	: SVC (I0CS);
0026	1F	02	1F	02	0008	0001	00	01	: RESTORE (RL);
0027	1F	02	1F	02	0008	0001	00	01	: SVC (BACK);
0028	1F	02	1F	02	0008	0001	00	01	: ERB3;
0029	1F	02	1F	02	0008	0001	00	01	: GOTO ERB3;
0030	1F	02	1F	02	0009	0001	00	01	: END.

FIN DE COMPILATION 0000 ERREUR(S) 0017 MOTS

#MSG -E D I T I O N - D E - L I E N S --
#/CALL EUIE
#/BI RAL.TOVE
#/ILNK

MONITEUR BACKGROUND

**
**MODULE NO 01
**

0072 TOVER
#/BI B1.TOVE
#/BLNK TOVER1

MONITEUR BACKGROUND

*
**
**MODULE NU 02
**

009A TOVER1
#/BI 02.TOVE
#/BLNK TOVER2

MONITEUR BACKGROUND

**
**MODULE NU 03
**

0093 TOVER2
#/B# 83.TGVE
#/BLNK TOVER3



MONITEUR BACKGROUND

**
**MODULE NO 04
**

0091 TOVER3
#/RLNK
#/ELNK

MONITEUR BACKGROUND

```
#/MSG -B U I L D - E N - M O D E - E S C L A V E - S U R - D 2 --
#/CALL BUILD
#/SLED
TOVER '006F
KIOV '0004
CUMIOV '0054
LLOIOV '005F
TLVER1 '0099
LVERB1 '0089
TOVER2 '0092
LVERB2 '008A
TOVER3 '0090
LVERB3 '0088

TASK S,'19
BEGIN '0000
END '0085
RUN '0072
#/CATAL IM,TOVER-AA
#/MSG -LA TAILLE DE L'IMAGE MEMOIRE,CALCULEE A PARTIR DES MESSAGES
#/MSG -BEGIN ET END,DONNE LA TAILLE MINIMALE DE LA PARTITION POUR TOVER
#/EL LP
#/MSG -L A N C E M E N T - E N - B A C K G R O U N D --
#/RUN TOVER-AA
```

MONITEUR BACKGROUND

ESSAI OVERLAY

#/EQJ

3 - BIBLIOTHEQUE TEMPS REEL

3.1 - INTRODUCTION

Les sous-programmes de la bibliothèque BIBRTE-RT ont pour but d'établir l'interface entre un programme utilisateur écrit en langage FORTRAN ou PL16 et les requêtes programmées du système RTES-D.

Une requête programmée est constituée d'une séquence d'instructions et en règle générale d'une table de paramètres qui spécifient l'opération demandée. Le sous-programme récupère les paramètres passés lors de l'appel (CALL), les met en forme pour que ceux-ci soient assimilables par la requête.

3.2 - SYNTAXE

Elle se présente sous la forme suivante :

```
CALL RNAME (P1, P2, P3,..., Pn)
```

où RNAME est un nom de sous-programme externe faisant partie d'une bibliothèque système et intégré au programme utilisateur par éditions de liens.

P1, P2, P3,..., Pn sont des adresses de variables ou de tableaux entiers.

3.2.1 - Syntaxe FORTRAN

Les requêtes proposées dans le projet de norme FORTRAN Temps réel (communication de l'instrument Society of America ISA-S61) sont traitées par RTES-D, ce sont par exemple :

```
CALL START (i, j, k, m)
```

```
CALL TRNON (i, j, k)
```

```
CALL WAIT (i, j, k)
```

3.2.2 - Syntaxe PL16

En langage PL16, les règles à observer sont les suivantes :

- Une tâche écrite en PL16 peut appeler une requête avec la syntaxe FORTRAN à condition que les divers paramètres de cette requête soient des pointeurs, (les paramètres passés en FORTRAN étant des adresses).
- La liste des paramètres doit être suivie d'une instruction de chargement du registre accumulateur avec le nombre de paramètres effectivement passés.

Exemple :

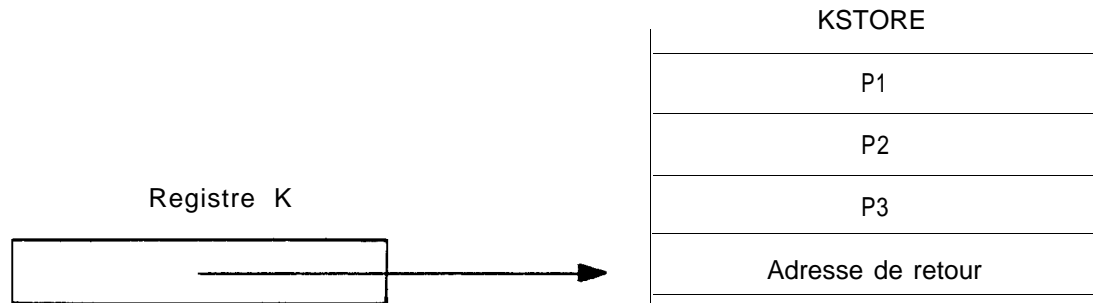
```
CALL RUN (@ NT, @ IERR, @ IPAR, RA := 3) ;  
CALL CEXIT ( RA := 0) ;
```

3.3 - PASSAGE DES PARAMETRES

En langage FORTRAN comme en langage PL16, le passage des paramètres s'effectue dans la KSTORE. Cette fonction est réalisée par les compilateurs.

Exemple :

```
CALL Spi (P1, P2, P3)
```



Important : Dans tous les cas, les paramètres passés sont des adresses de variables ou de tableaux.

3.4 - FONCTIONS REALISEES PAR UN SOUS-PROGRAMME

1. Récupération des paramètres dans la KSTORE, élaboration d'une table de paramètres (RPB) pour la requête (si celle-ci en comporte).
2. Pré-détection d'erreurs dans la liste des paramètres, dans certains cas.
3. Tassage de la KSTORE.
4. Réalisation du branchement vers le sous-programme superviseur par l'intermédiaire de l'instruction SVC de numéro approprié.
5. Retour à l'appelant.

Nota : La table des paramètres d'une requête est interne au sous-programme.

3.5 - PRE-DETECTION D'ERREURS DANS UN SOUS-PROGRAMME

Il y a deux cas de pré-détection d'erreurs dans la bibliothèque des sous-programmes :

3.5.1 - Requetes sur liste d'événements :

Le numéro de classe pour au moins deux événements est différent.

3.5.2 - Liste des paramètres incomplète :

Dans le cas où les requêtes comportent un nombre imposé de paramètres, le nombre de paramètres transmis étant inférieur à celui-ci.

Remarque : Dans le cas où le nombre de paramètres transmis est supérieur au nombre imposé la requête est acceptée, les paramètres étant exploités dans l'ordre où ils sont déclarés.

3.6 - COMMENT EXPLOITER LE COMPTE RENDU DE REQUETE

3.6.1 - Requête s'étant déroulée en tout ou partie :

a) Langage FORTRAN :

Le compte rendu de traitement est rangé à l'adresse précisée lors de l'appel.

b) Langage PL16 :

Identique au langage FORTRAN, cependant la valeur du compte rendu est également contenue dans le registre accumulateur.

3.6.2 - Erreurs détectées par le sous-programme

La requête n'est pas effectuée, l'appelant n'est pas suspendu et le compte rendu de traitement n'est pas rangé.

a) Langage FORTRAN :

Il est conseillé avant d'effectuer l'appel d'initialiser la variable chargée de recevoir le compte rendu avec une valeur non utilisée par les requêtes (- 1 par exemple) - si après un appel cette valeur n'est pas modifiée, c'est qu'une erreur a été détectée par le sous-programme.

b) Langage PL16 :

En retour de l'appel, le registre accumulateur a pour valeur 6.

3.7 - COMPOSITION DE LA BIBLIOTHEQUE

La liste des sous-programmes composant la bibliothèque est décrite en annexe. En règle générale, elle se compose des sous-programmes des requêtes du système RTES-D, à l'exception de :

- IOCS
- FMS
- BCHLR
- BACK
- BACKR

Outre ces sous-programmes, la bibliothèque comporte plusieurs sous-programmes communs. Ils rendent ainsi plus rationnelle l'occupation mémoire puisqu'ils ne sont pas dupliqués lors de leur utilisation. Une telle organisation interdit l'utilisation dans les programmes écrits en langage PL16 ou FORTRAN du symbole $RTESD_i$, $1 \leq i \leq 7$.
(RTESDi : nom de ces sous-programmes)

3.8 - MOTS RESERVES DU LANGAGE PL16

Trois mots réservés du langage PL16 : EXIT, START et WAIT sont utilisés par RTES-D pour la reconnaissance des requêtes programmées de même nom. Deux d'entre eux, START et WAIT, sont imposés par la norme FORTRAN temps réel, le troisième n'est pas normalisé mais couramment utilisé (fin de tâche). Ces trois sous-programmes seront employés sous des noms différents :

- WAIT, START, EXIT en FORTRAN
- CWAIT, CSTART, CEXIT en PL16.

3.9 - UTILISATION DE LA BIBLIOTHEQUE

3.9.1 - Mise en oeuvre :

La bibliothèque de sous-programme est un fichier indexé contenu dans la E2 de la cartouche de génération. Son nom est BIBRTE-RT.

3.9.2 - Editions de liens d'un programme

Dans l'exemple 1 d'utilisation donné en annexe, la tâche écrite en PL16 de numéro d'appel 12 a pour priorité 12 ; elle sera "buildée" en mode esclave. Cette tâche active, par la requête RUN, la tâche de numéro d'appel 11 avec le paramètre 1.

Dans l'exemple 2, la tâche possède les mêmes attributs mais est écrite en FORTRAN.

Elle réalise les fonctions suivantes :

- Activation immédiate de la tâche de numéro d'appel 11 avec le paramètre 2 (RUN)
- Suspension pendant 1 minute (WAIT)
- Activation différée et périodique de la tâche de numéro d'appel 11 avec le paramètre 2 (START).

3.10 - FONCTIONNEMENT DE LA BIBLIOTHEQUE

Tous les sous-programmes et toutes les requêtes restituent tous les registres de la machine à l'exclusion des registres index et accumulateur (qui contient la valeur du compte rendu de traitement). Une exception est faite pour le registre B qui est détruit par le sous-programme EXIT (CEXIT en langage PL16).

MNEMONIQUE	TAILLE en Mots	Prof. KSTORE	RTESD 1	RTESD 2	RTESD 3	RTESD 4	RTESD 5	RTESD 6	RTESD 7
RUN	39	30	X	X	X	X	X		
CSTART	58	30	X	X	X	X	X		
STARTW	49	30	X	X	X	X	X		
TRNON	74	30	X	X	X	X	X		
TRNONW	64	30	X	X	X	X	X		
CWAIT	40	30	X		X	X	X		
DATAG	31	30	X		X	X			
CEXIT	21	30 (1)							
OFF	35	30	X		X	X	X		
INHIB	35	30	X		X		X		
WEVENT	36	30	X		X	X	X		
WEVAND	39	30	X	X	X	X		X	
WEVOR	67	30	X	X	X	X		X	
SEVENT	33	30	X		X	X	X		
SEVDEL	37	30	X		X	X	X		
OFFDEL	35	30	X		X		X		
REVENT	33	30	X		X	X	X		
TEVENT	35	30	X		X	X	X		
REDGET	41	30	X		X	X	X		
REDPUT	41	30	X		X	X	X		
RESDEF	37	30	X		X	X	X		

(1) Pour une tâche non résidente avec OVERLAY, la profondeur exigée est plus importante = 64 mots.

MNEMONIQUE	TAILLE en Mots	Prof. KSTORE	RTESD 1	RTESD 2	RTESD 3	RTESD 4	RTESD 5	RTESD 6	RTESD 7
RRQST	33	30	X		X	X	X		
RRLSE	33	30	X		X	X	X		
STADEF	33	30	X		X	X	X		
STAGET	37	30	X		X	X	X		
TIME	32	30	X		X	X			
RWAIT	33	30	X		X	X	X		
RACT	39	30	X		X	X	X		
WAIT	28	30	X		X	X			
EXIT	19	30							
START	47	30	X	X	X	X	X		
TCALL	57	30	X	X	X	X	X		X
TCALLW	65	30	X	X	X	X	X		X
SEND	40	30	X		X	X	X		
RECEIVE	35	30	X		X	X	X		
WTIME	34	30	*		X	X			
COMAND	34	30	X		X	X			

Sous-programmes	TAILLE en Mots
RTESD 1	13
RTESD 2	11
RTESD 3	18
RTESD 4	16
RTESD 5	18
RTESD 6	36
RTESD 7	26

Exemple d'évaluation d'occupation mémoire

Soit une tâche effectuant les requêtes RUN, DATAG et CEXIT ; l'occupation mémoire des sous-programmes "ink-édités" se répartit ainsi :

RUN = 39 mots
DATAG = 31
CEXIT = 21

auxquels il convient d'ajouter les sous-programmes

RTESD1 = 13 mots
RTESD2 = 11
RTESD3 = 18
RTESD4 = 16
RTESD5 = 18
Total = 167 mots

4 - UTILISATION DES DISQUES SOUS LES SYSTEMES DISQUES STANDARD

4.1 - RAPPELS

4.1.1 - Caractéristiques techniques des disques à têtes fixes

Modules FHD01A-0

- capacité : 512 K mots répartis en 128 pistes à 32 secteurs de 128 mots (+1 mot de contrôle)
 - cadence d'échange : 230 K mots/s
 - temps d'accès moyen : 10 ms
 - utilisation : mode canal HDC
- Un coupleur ne contrôle qu'un disque.

4.1.2 - Caractéristiques techniques des disques à tête mobile

Modules CDB10-0

- capacité : 10 Moctets organisés en deux disques de 400 cylindres de 48 secteurs de 128 mots (+2 mots de contrôle)
chaque unité comporte un disque fixe de 5 Moctets et un disque amovible de même capacité
- cadence d'échange : 156 K mots/s
- temps d'accès : cylindre à cylindre 10 ms
à un cylindre moyen 35 ms
maxi 70 ms
- temps d'accès moyen en rotation : 12,5 ms
- utilisation : mode canal MDC ou HDC

Le coupleur peut gérer jusqu'à 4 unités de disques.

La première unité est raccordée au coupleur, les suivantes sont reliées par chaînage sur l'unité précédente.

4.1.3 - Notion de volume, d'espace et d'unité fonctionnelle disque

On appelle volume un disque amovible identifié par un label.

On appelle espace disque une partie du volume définie par son adresse physique (secteur 0 d'un cylindre) et sa longueur (nombre entier de cylindres).

Un volume peut contenir au maximum 16 espaces numérotés de 0 à 15.

On définit au moment de la génération du système pour chaque unité disque une certaine découpe caractérisée par le nombre de portions de disque. Chaque portion s'appelle unité fonctionnelle disque.

La première de ces FU sert à repérer l'unité physique.

Les adresses et les longueurs des autres FU peuvent être fixées :

- à la génération du système (Macros de Genio),
- ou dynamiquement par des commandes opérateur au moment du montage du volume sur l'unité physique.

4.2 - UTILISATION DE IOCS

IOCS assure la gestion des F.U disques.

- lecture ou écriture d'une zone à partir d'une adresse de secteur
- vérification des écritures par la lecture de contrôle
- vérification de la validité de demande par rapport aux paramètres des F.U.
- protection en écriture
- positionnement du bras sur un cylindre (disque à tête mobile).

De plus, pour les disques à tête mobile :

- toute demande peut donner lieu à un mouvement
- un échange peut donner lieu à un mouvement intermédiaire.

Les échanges sont limités à 8K mots pour les disques à têtes fixes, à 6K mots pour les disques à tête mobile.

4.3 - UTILISATION DE FMS

FMS est un utilisateur de IOCS comme les autres. Il travaille sur des FU qui lui ont été attribuées à la génération du système ou après les commandes de montage de volume.

Des fichiers appartenant à des utilisateurs différents sont implantés dans la même FU.

FMS réalise pour un utilisateur, la création, l'utilisation et la gestion des fichiers.

De plus, il assure la protection d'un fichier vis à vis des autres utilisateurs de FMS travaillant sur la même FU.

Un utilisateur de FMS ne peut donc avoir accès qu'aux FU gérées par FMS.

Mais en utilisant directement IOCS, on a accès à toutes les FU. Aucune vérification sur la validité d'accès à une FU par un utilisateur de IOCS n'est réalisée :

Il est donc très recommandé de n'utiliser que FMS pour traiter des informations sur disque (sûreté et facilité d'utilisation).

4.4 - EXPLOITATION SOUS LES SYSTEMES DISQUES STANDARD

Les systèmes disques standard utilisent des FU gérées directement par IOCS (zones systèmes) et des FU gérées par FMS (fichiers systèmes, tâches résidentes, non résidentes).

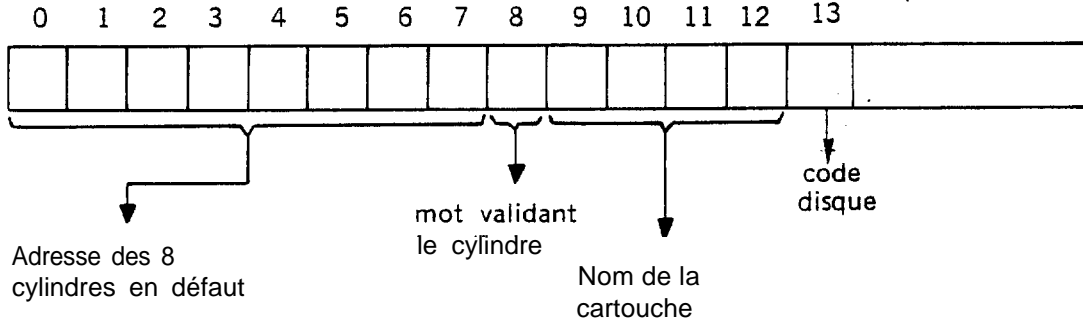
On a donc coexistence de FU gérées par IOCS et par FMS. Les utilisateurs disposent de FU gérées par FMS ; l'utilisation de FU gérées par IOCS pour un utilisateur, devrait être très rare puisque FMS répond en règle générale aux besoins de l'utilisateur.

4.5 - PRINCIPE D'UTILISATION DES DISQUES A TETE MOBILE

4.5.1 - Initialisation du disque

Les disques à cartouches (ainsi que les plateaux fixes) doivent être initialisés. Le programme d'initialisation réalisera la première écriture et la recherche des cylindres en défaut. Il créera dans le cylindre 0 le secteur 3 avec le nom du disque (fixe ou cartouche) et la liste des cylindres en défaut.

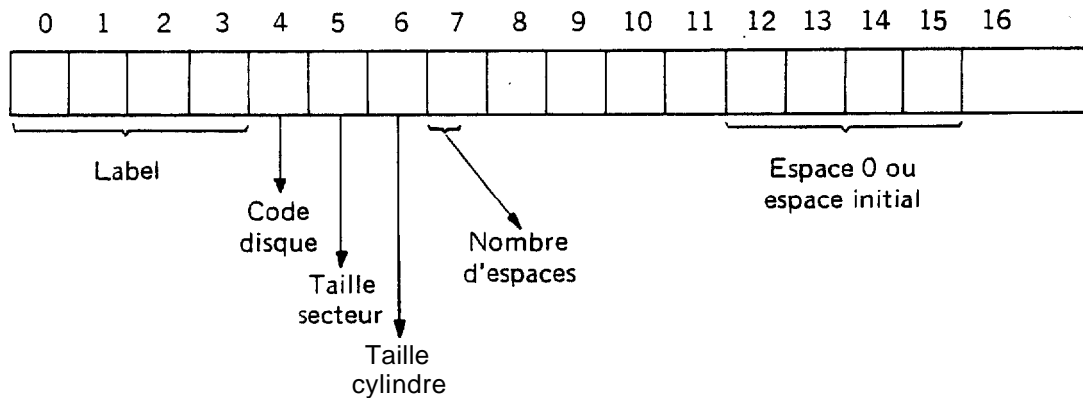
Structure du secteur 3 du cylindre 0



Le nom est constitué de sept caractères au plus suivi de **CR** qui sert d'arrêt sur le nom.

Le processeur FUP4 permet de définir une découpe du volume en espace (commande SDEF). Le secteur 4 du cylindre 0 constitue la table de structure du volume.

Structure du secteur 4 du cylindre 0



La description complète des secteurs 3 et 4 est faite dans le manuel IOCS.

4.5.2 - Opération de montage de volume

Lorsqu'on monte un volume sur une unité physique l'opérateur doit, à l'aide des commandes de montage de volume, déclarer au système la structure du volume. L'action réalisée par ces commandes est de vérifier le nom de la cartouche, de réaliser l'affectation SU-ESPACE, de charger dans les tables d'IOCS les adresses des cylindres en défaut.

Pour le plateau fixe, les cylindres en défaut sont déclarés à IOCS au moment du lancement du système 1 de RTES-D. Dans le cas où huit cylindres sont en défaut le système émet alors le message :

"DISQUE HORS SERVICE"

Une intervention de dépannage est alors nécessaire.

4.6 - GESTION DES CYLINDRES EN DEFAUT

A chaque demande d'accès sur le cylindre j , le "driver" transforme j en $j + k$ si k est le nombre de cylindres en défaut d'adresse inférieure ou égale à celle mise à jour.

Exemple :

cylindre demandé 40
cylindres en défaut 1, 5, 10, 43, 50
le cylindre adressé sera 44.

4.7 - CONFIGURATION EN FU DES DISQUES

Les systèmes BOS-D et RTES-D sont nécessairement sur une FU disque particulière contenant la version courante du système configuré pour l'installation.

Sur une installation, on peut diviser l'ensemble des disques en deux catégories :

- 1 - les disques qui comportent la version du système qui sera utilisée (disque fixe à tête mobile ou fixe, ou cartouche) ; ceux-ci peuvent avoir une découpe en FU quelconque ; le système porte la configuration du disque sur lequel il se trouve,
- 2 - les disques fixes (à tête mobile ou fixe) : leurs configurations sont décrites dans le système.

Les FU gérées par IOCS et FMS (option grand disque seulement) peuvent commencer à partir du cylindre 0. Mais l'utilisateur de cette FU doit tenir compte des secteurs 0 à 4 qui ne sont pas protégés.

5 - GENERATION D'UN SYSTEME RTES-D

5.1 - PRINCIPE

- Elle s'opère sous les systèmes BOS-G ou BOS-D grâce au processeur GENRTD.
- La génération d'un système RTES-D consiste à "link-éditer" IOCS, FMS configurés pour les besoins de l'application ainsi que les "drivers" des périphériques de l'installation aux M.O.L. (Modules Objet Link-éditables) de RTES-D.
 - . La cartouche de génération contient les M.O.L. de RTES-D sous la forme d'un fichier indexé "RTES D-MO".
 - . La génération s'opère de la façon suivante :

5.2 - GENIO CONFIGURATION DE IOCS

Le moniteur d'entrées-sorties IOCS de RTES-D doit être configuré en fonction des périphériques de l'installation et du découpage désiré des disques en unités fonctionnelles.

L'ensemble des macro-instructions de GENIO (cf. Manuel d'utilisation de IOCS) permet de décrire les périphériques de cette installation.

Il faut donc écrire un jeu de macro-instructions sur le support de son choix (cartes, ruban perforé, fichier disque...).

Ces macro-instructions seront traitées automatiquement lors de la génération proprement dite (voir § 5.4).

Exemple :

Soit une installation comprenant :

- un téléimprimeur
- une unité de disques avec plateau fixe et cartouche
- un lecteur de cartes et une imprimante
- un lecteur et un perforateur de rubans
- une horloge temps réel d'une fréquence de 50 Hz.

Tous ces périphériques étant aux adresses débanalisées, les macro-instructions à écrire sont les suivantes :

```
%RTESD
%FSMON GVOL
%NIVEAU 13
%HTR SNIV=0  FREQ=50
%NIVEAU 14
%PUCD SNIV=1  MODE=HDC  ADR='30  ITN=1  IOP=  NBV=1
%FUICD 19  VOIE=0  FIXE=y
%FUESPCD 13
%FUESPCD 14
%FUESPCD 18
%FUICD 36  VOIE=0  FIXE=N
%FUESPCD 15
%FUESPCD 16
%FUESPCD 17
%CR
%LP
%NIVEAU 15
%TTY
%HR
%HP
%ENDGEN
*END
```

Remarques : Plusieurs contraintes doivent être respectées pour la configuration de IOCS de RTES-D.

- . La 1ère macro doit être %RTES-D.
- . La gestion de volume est obligatoire sur RTES-D, aussi faut-il faire suivre la macro %RRTES-D de la macro %FSMON pour introduire les fonctions spéciales moniteur DMON, MONT, READ.
- . Les unités fonctionnelles TS, TK, D1 et D2 (numéros 2, 3, 13 et 14) doivent obligatoirement être définies.
- . Les FU D1 et D2, dans le cas d'un disque à têtes mobiles, sont obligatoirement sur l'unité de numéro 0 (paramètre VOIE = 0 dans la macro-instruction FUICD précédente).
- . Dans le cas où l'on dispose d'un périphérique de dialogue réservé au "background" (téléimprimeur ou console de visualisation alphanumérique), il faut définir les FU F7 (27) en sortie et F8 (28) en entrée sur ce périphérique.
- . La macro-instruction HTR ne doit pas être omise. Elle permet de configurer l'horloge temps réel obligatoire dans la configuration matériel de RTES-D. La fréquence de cette horloge ne doit pas dépasser 1000 Hz. La gestion de l'horloge est assurée par le "driver" DRVHTR contenu dans la bibliothèque des "drivers" BIBDRV.
- . Les fonctions spéciales de montage de volume sont dans la bibliothèque BIBVOL-:S.

5.3 - GENFMS CONFIGURATION DE FMS

5.3.1 - Les macro-instructions de GENFMS décrivent le moniteur de gestion de fichiers FMS (cf. manuel d'utilisation de FMS)

Elles permettent :

- de définir les FU disques initiales et les zones mémoire nécessaires à la gestion des FU gérées par FMS,
- de générer les commandes nécessaires à l'édition de liens de FMS.

5.3.2 - Configuration d'une FU disque gérée par FMS

Pour IOCS, les noms des FU disque sont définis par GENIO au moment de la génération du système. Leurs caractéristiques sont transmises à IOCS au moment du montage de volume.

Pour que FMS puisse gérer une FU disque, il faut :

- réserver une zone mémoire permettant la gestion de la FU,
- initialiser le support physique de la FU.

Première étape

Elle est réalisée à la génération du système au moyen de GENFMS. Elle permet à l'aide de la macro %PUFMS de définir la FU initiale et la place que l'on doit réserver pour tous les descripteurs des FU FMS de l'unité physique, afin de contenir les tables d'allocation des granules (TAG) de chaque FU FMS.

Les unités physiques FMS générées sont vides, c'est-à-dire que les FU correspondantes sont inexistantes pour FMS.

Leur existence sera réalisée lors du montage de volume.

Deuxième étape

Cette étape consiste :

- à initialiser le support physique des FU gérées par IOCS et FMS, c'est-à-dire à structurer le volume en décrivant l'adresse de début et la longueur de chaque espace,
- à initialiser une zone disque située au début de la FU par un certain nombre de renseignements tels que la taille des granules et l'emplacement des granules libres.

La définition de structure et l'initialisation sont réalisées par les commandes SDEF et FUINI du processeur FUP4 (cf. manuel de référence FUP).

Règles d'utilisation de FUINI

organisation standard	3 secteurs ≤ taille des granules ≤ 256 secteurs
	2 ≤ nombre de granules ≤ 2000
organisation grand disque	3 secteurs ≤ taille des granules ≤ 32 K - 1 secteurs
	1 ≤ nombre de granules ≤ 32 656

nombre de granules x taille des granules ≤ taille de la FU (en secteurs).

Règles d'utilisation de SDEF

FU	désigne la FU initiale de l'unité physique sur laquelle est monté le volume
0 ≤	numéro d'espace ≤ 15
0 ≤	adresse en cylindre du début FU < longueur du disque
	longueur en cylindre de la FU telle que 0 ≤ longueur + adresse < longueur du disque
	0 ≤ longueur table d'allocation des granules ≤ 125

L'espace correspondant à D1 doit obligatoirement commencer au cylindre 0.

Exemple : Opération à faire sous BOS-D

Définition de la structure d'une cartouche

```
CALL FUP4
IDEF, D9, LABEL, 1, 128, 48, 400
DMONT D9
SDEF, D9, LABEL, 1, 1, 339
SDEF, D9, LABEL, 2, 1, 300
SDEF, D9, LABEL, 3, 301, 399
```

Remarque :

IDEF permet de définir l'espace initial. Cette fonction est faite aussi par le programme de formatage.

SDEF permet de définir les autres espaces et de mémoriser le découpage dans la table d'espace du disque.

Initialisation de l'espace 1 géré par FMS :

```
MONT D9
CALL FUP4
SUSP U1, 1
FUINI, U1, 16, 1197
```

Les procédures à employer pour rendre compatible les anciens supports disque sont expliquées dans le manuel de référence FUP chapitre FUP4.

Troisième étape

Elle consiste à configurer dynamiquement les tables d'IOCS et de FMS par les informations inscrites sur le support.

Cette reconfiguration se fait lors de :

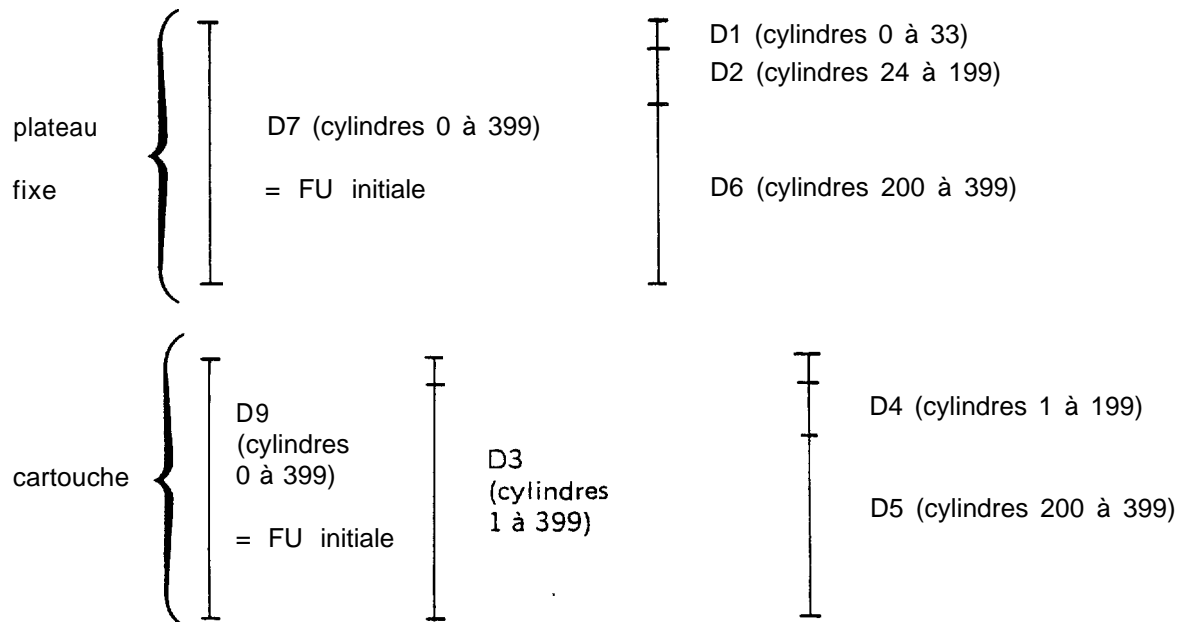
- l'initialisation du système pour l'unité physique (PU) système, celle contenant D1 et D2,
- par la commande MONT, fu initiale pour les autres unités physiques.

Exemple :

```
MONT, D9
LABEL : LABEL
SPACE 0  FU D9
SPACE 1  FU D3
SPACE 2  FU D5
SPACE 3  FU D6
```

5.3.3 Exemple de macro-instructions de GENFMS

L'installation configurée au § 5.2 comporte une unité de disque à têtes mobiles dont le découpage en FU disque est le suivant :



Les macro-instructions de GENFMS à écrire pour générer les tables de FMS correspondant à cette découpe en FU et PU :

```
%PUFMS FUI = D7 OFU = 0 NBFMSMAX = 2 PUNBGRAN = 1000
%PUFMS FUI = D9 OFU = 0 NBFMSMAX = 3 PUNBGRAN = 3500
%SEQ
%IND
%DIR
%BUF
%ADR
%ALACARTE
*END
```

5.4 - GENRTD GENERATION D'UN SYSTEME RTES-D

GENRTD permet de générer l'ensemble des commandes nécessaires à la génération de RTES-D. La description des différentes phases de la génération est faite à l'aide de macro-instructions de GENRTD.

Les différentes phases de la génération d'un système RTES-D sont :

- configuration de IOCS
- configuration de FMS
- édition de liens des modules composant le système (IOCS, FMS, "drivers" des périphériques de l'installation et enfin le M.O.L. de RTES-D),
- création de l'image mémoire du système,
- implantation dans la FU D1 (commandée par la macro %SYSTEM).

Les macro-instructions communes à la génération des systèmes BOS-D et RTES-D ne seront pas détaillées ici, on les trouvera décrites au § 7.4.3 du manuel de référence de BOS-D.

Liste des macro-instructions communes :

%FUGENE unité fonctionnelle disque

%GENIO MACRO ON { nomfic - catg [, FU] }
 HR
 CR
 TK }

%NOYIOCS M2

%GENFMS MACRO ON { nomfic - catg [, FU] }
 HR
 CR
 TK }

%FMSG

%BIBDRV { nomfic - catg [, FU] }
 *
%DRV ON { nomfic - catg [, FU] }
 HR }

%FIN

5.4.1 - Macro-instructions spécifiques de la génération d'un système RTES-D

```
%LINK RTESD
```

But : édition de liens de IOCS, FMS et RTES-D.

```
%BUILD RTESD { AVEC } BACKGROUND [ NAMSYS { nom } ] [ FIN RACINE { HHHH } ]
               { SANS }
```

But : création du fichier image mémoire de RTES-D.

Paramètre :

- AVEC ou SANS BACKGROUND sert à préciser la version de RTES-D (gestion ou non du background).
- NAMSYS : s'il est omis ou s'il est égal à * l'image mémoire sera cataloguée sous le nom RTES-D - : S.
Sinon sous le nom : nom - : S
- FIN RACINE : S'il est omis ou s'il est égal à * l'adresse de fin d'implantation sera égale à FF80.
Sinon elle sera égale à la valeur donnée en paramètre.

5.4.2 - Remarque concernant l'utilisation de la macro-instruction

```
%SYSTEM { 1 } NUMSECT adresse
         { 7 }
```

Lors de la génération d'un système RTES-D le paramètre 1 (n° du système) peut prendre une valeur comprise entre 1 et 7.

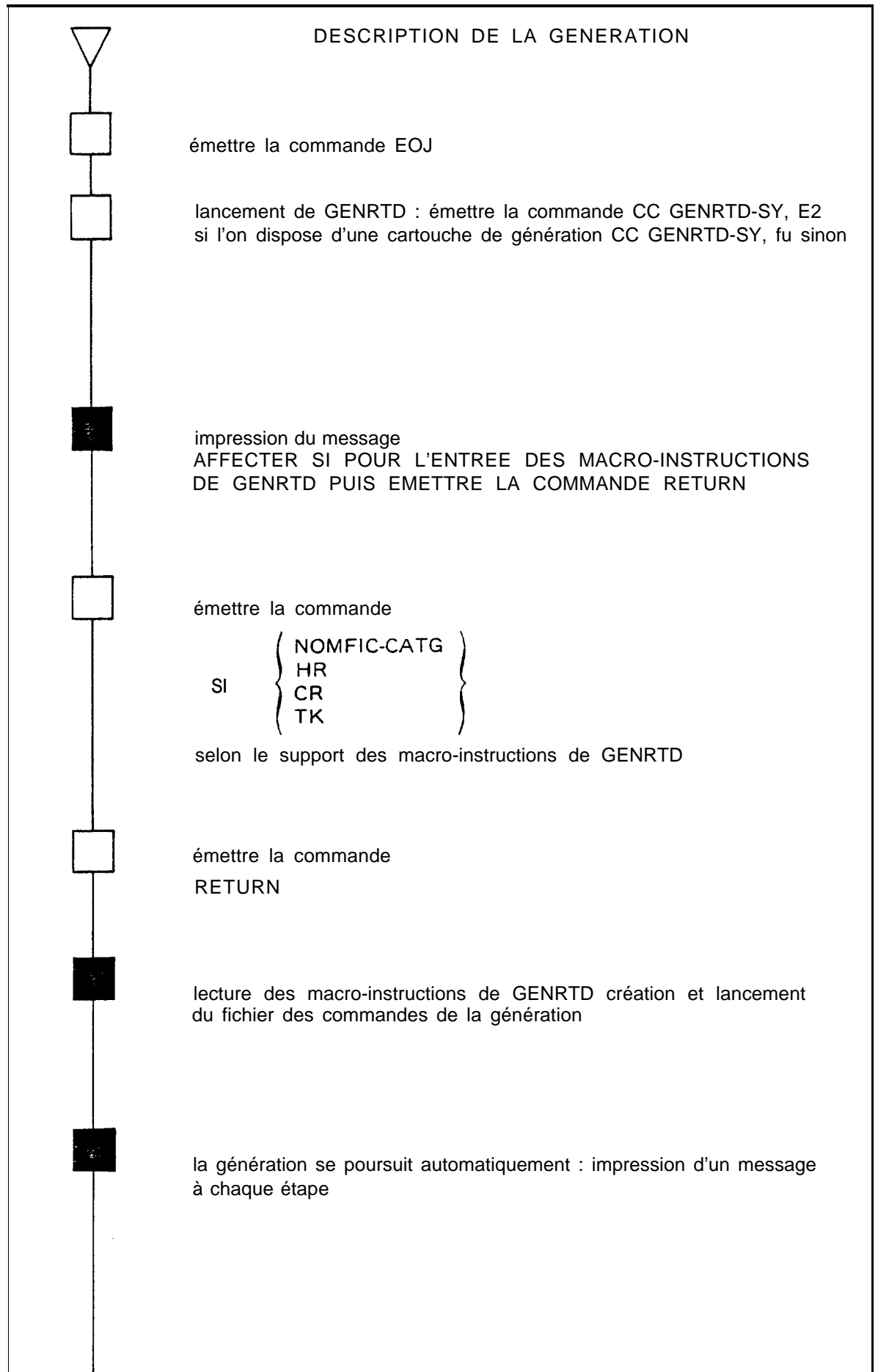
5.4.3 - Exemple de génération de RTES-D

Macro-instructions de "GENCOM"

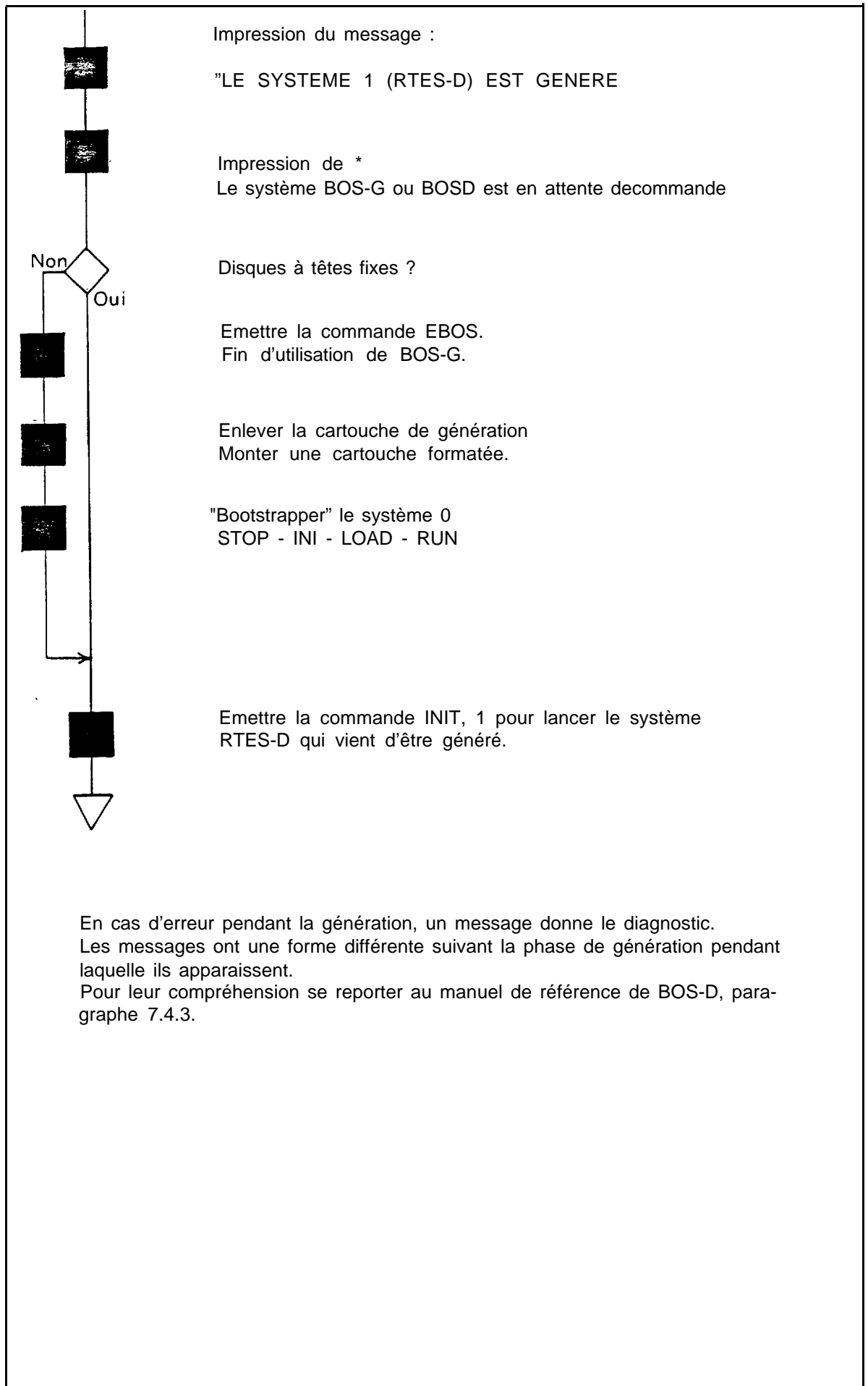
```
%GENIO MACRO ON CR
%NOYIOCS M2
%GENFMS MACRO ON CR
%LINK RTESD
%BIBDRV *
%BUILD RTES-D AVEC BACKGROUND
%SYSTEM 1 NUMSECT 1000
%FIN
* END
```

REMARQUE :

Si l'on désire utiliser la même configuration pour FMS que celle utilisée pour BOS-D, les macro-instructions de GENFMS sont inutiles, les fichiers qu'elles génèrent étant déjà créés.



Solar 16



6 - UTILISATION DU MONITEUR "BACKGROUND"

Le moniteur "background" BACKM est une tâche de RTES-D dont le but est d'apporter à l'utilisateur les services rendus par le système BOS-D.

En particulier, BACKM permet la production et la mise au point de tâches temps réel en simultanéité avec le déroulement de l'application.

La description détaillée du langage de commande et des requêtes accessibles à un programme utilisateur sont fournies dans le manuel de référence de BOS-D. Seuls sont repris dans ce chapitre les points présentant une différence entre les deux moniteurs BOS-D et BACKM.

6.1 - LES REQUETES DE BACKM

Toutes les requêtes programmées du système BOS-D sont maintenues dans le moniteur "background" BACKM. Les conditions d'utilisation (en particulier, la profondeur nécessaire de la zone pointée par le registre K lors de chaque requête) sont identiques pour les deux systèmes.

Une seule différence : En standard, après toute demande d'entrée-sortie portant sur la zone mémoire occupée par les processeurs IOCS ne rend le contrôle que lorsque l'échange est effectivement terminé. En effet, le système analyse et transforme éventuellement chaque requête adressée à IOCS, l'opération d'entrée-sortie étant toujours demandée avec retour en fin d'échange. Ce processus, induit par le mécanisme de "swap", n'implique qu'une seule contrainte pour l'utilisateur : tout IOCB doit avoir une longueur d'au moins 5 mots, le mot (4) de l'IOCB étant utilisé par IOCS pour positionner un événement fin d'échange.

Lorsque la commande d'activation BACK, émise sous RTES-D, comporte le paramètre NOSWAP, toute requête adressée à IOCS est inchangée (les échanges se déroulent avec le mode de retour spécifié dans l'IOCB).

En standard, les requêtes spécifiques du temps réel sont inefficaces. Deux exceptions cependant : les requêtes EXIT et DATAG émises par un processeur utilisateur sous le contrôle du moniteur BACKM sont assimilées aux requêtes ABOS et RBOS de BOS-D.

Il est possible cependant de rendre toutes les requêtes spécifiques du temps réel accessibles au "background" (voir Manuel de référence de RTES-D, commande BACK).

6.2 - LES COMMANDES DE BACKM

Le moniteur indique qu'il est en attente d'une commande en écrivant sur le périphérique associé à l'unité symbolique "Listing Log" (LL) le message :

D'une manière générale les commandes que peut alors émettre l'utilisateur sont celles de BOS-D. Cette règle comporte cependant un certain nombre d'exceptions.

Ainsi les commandes suivantes ne sont pas reconnues par BACKM :

PASS, NPAV, FORE, SYST, TLOAD, PRUN, IRUN, MEMORY, PRINT,
CONF, ZCONF, commandes de configuration du système BOS-D.
INIT permettant le passage sous le contrôle d'un système quelconque.
IBOS, commande de réinitialisation de BOS-D.

La forme des commandes reconnues par BACKM est décrite dans le manuel de référence de BOS-D.
Deux commandes diffèrent quelque peu :

a) Commande JOB : cette commande comporte un paramètre supplémentaire. Sa forme est la suivante :

JOB $nom, catg, \begin{cases} SU \\ FU \end{cases},,, taille$ \textcircled{cr}
--

- nom** est un symbole alphanumérique de 6 caractères maximum servant à identifier le travail
- catg** est le nom de catalogue pris par défaut pour la recherche des fichiers nommés dans les commandes.
Par défaut ce nom est "null" c'est-à-dire celui d'un catalogue commun.
- FU-SU** est le nom de la FU ou SU disque (U1 à UF) où se trouve le catalogue précédent.
Par défaut on prend la FU comportant le fichier support de BACKM.
- taille** permet de spécifier le nombre de K mots requis pour l'exécution du "job" ; la limite maximale autorisée est 64 Kmots ; 4 Kmots sont requis par le système par défaut.

Le système allouant un nombre entier de partitions, le nombre de K mots occupés pour l'exécution du "job" sera donc supérieur ou égal au nombre demandé.

Le premier paramètre est seul obligatoire. Toutefois, le rôle de chaque paramètre est fonction du nombre de virgules qui le précèdent.

Exemple : JOB FTXAB, , D2 Catalogue "null"
 4 K mots alloués par le système

b) Commande OPTION : intégration au système des modules optionnels.

La commande OPTION permet d'intégrer au moniteur BACKM des modules implantés en format binaire translatable dans la bibliothèque système BIBACK. Cette bibliothèque contient en standard un seul module : la "bufferisation" des entrées-sorties (BUFFER). Cependant les modules suivants peuvent être intégrés à BIBACK :

- Flottant programmé double précision (DFLOAT)
- Outil de mise au point (DRIP16)
- Floppy Disk Management Monitor (FDM).

Chaque module est alimenté dans la partition du moniteur BACKM, immédiatement après le système. Certains modules (le module BUFFER par exemple) comportent deux parties :

- séquence d'initialisation dont le but est la mise à jour du système (actualisation des tables, communication des requêtes introduites par le module.....).
- option proprement dite qui, après l'initialisation, fait partie intégrante du système et ceci durant toute la vacation "background" (la place de la séquence d'initialisation est récupérée par le système).

- Flottant programmé double précision

DFLOAT doit être intégré dans l'article : SDFLOAT de la bibliothèque BIBACK - : S au moyen de l'utilitaire FUP6.

L'opération de chargement et lancement est réalisée automatiquement par la commande :

OPTION DFLOAT 

- Outil de mise au point DRIP16

DRIP16 doit être intégré dans l'article : SDRIP16 de la bibliothèque BIBACK - : S au moyen de l'utilitaire FUP6.

L'opération de chargement et lancement est réalisée par la commande :

OPTION DRIP16 

6.3 - LES PROCESSEURS DE BACKM

Chaque vacation "background" débute après émission sur le dispositif de dialogue "foreground" d'une commande BACK comportant en paramètre le numéro de la première partition utilisable par les processeurs.

D'autre part, la commande INIT permet le passage du contrôle de BOS-D à RTES-D et vice versa.

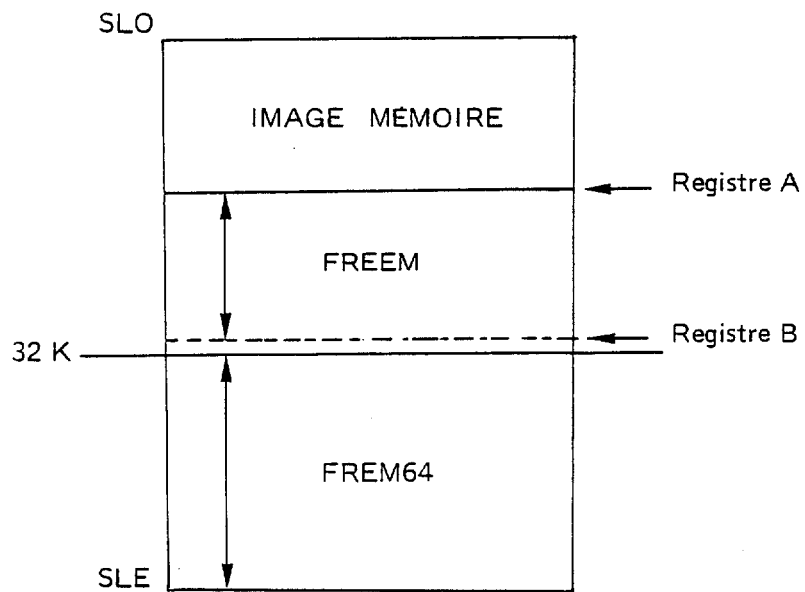
De manière à minimiser l'encombrement sur disque un même processeur doit donc pouvoir s'exécuter à des implantations mémoire différentes. Ceci n'est possible que si le processeur se déroule en mode esclave (intégration au moyen de la commande SLOD du "builder"). Le processeur est alimenté en mémoire dans les partitions allouées à Backm, à partir de l'adresse indiquée dans le mot 0 du descripteur de l'image mémoire. Cette adresse doit être soit nulle soit supérieure à 50.

Cependant, certains processeurs ne peuvent s'exécuter qu'en mode maître. De manière à pouvoir les utiliser sous le contrôle de BACKM l'utilisateur doit donc les intégrer au moyen de la commande MLOD du "builder" comportant en paramètre l'adresse de la première partition où ils doivent s'exécuter.

Acquisition des limites de la zone disponible

La requête FREEM permet d'obtenir les adresses relatives à SLO, début et fin de la mémoire libre dans les registres A et B.

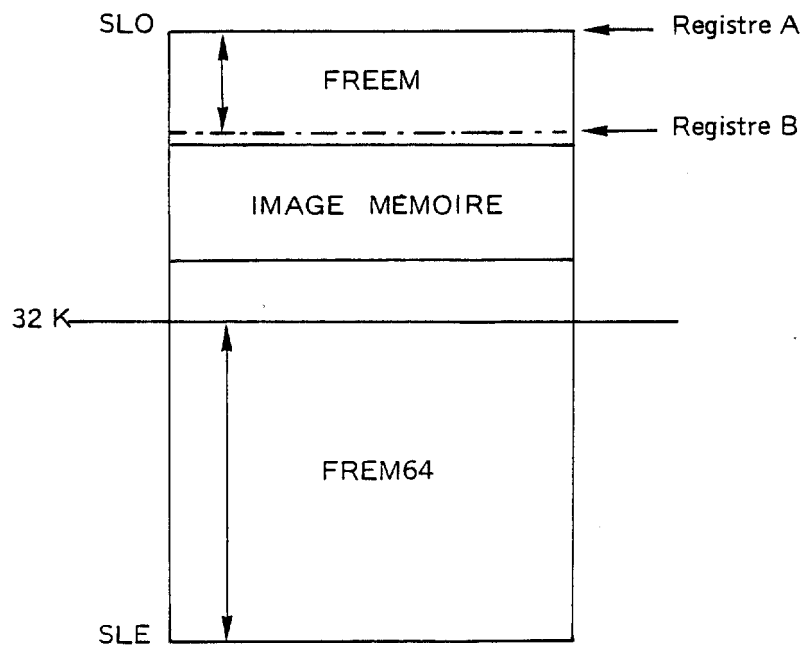
Exemple 1 - Cas adresse d'implantation égale à zéro



Remarques :

- La zone libre se situe derrière l'image mémoire.
- L'adresse de fin est limitée à 32K - 51.
- Si l'image mémoire est supérieure à 32K mots on a zéro dans les registres A et B.

Exemple 2 - Cas adresse d'implantation différente de zéro



Remarques :

- La zone libre se trouve devant l'image mémoire.
- On a dans le registre A zéro et dans B l'adresse d'implantation - 51.

La requête FREM64 (' 1A) permet d'obtenir dans les registres A et B les limites de la zone disponible située au delà des 32 K.

En ce qui concerne les images mémoires buildées en mode maître, la requête FREEM donne dans les registres A et B les adresses absolues de début et fin de la zone libre.

Sous BACKM comme sous BOS-D, les deux commandes RUN et LOAD permettent la mise en oeuvre des processeurs utilitaires suivants :

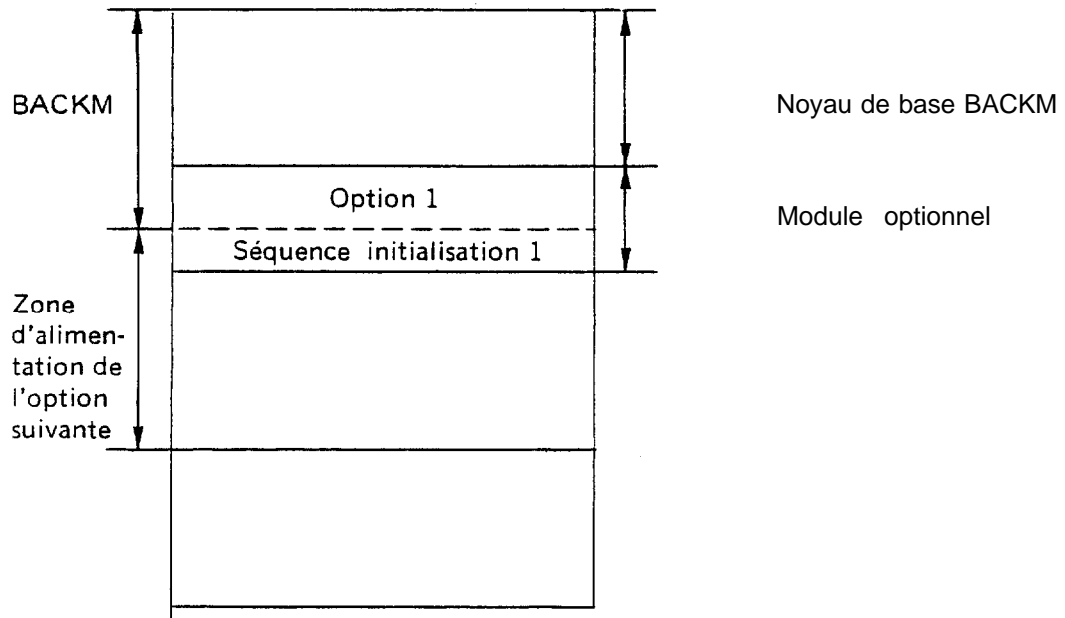
- module DRIP16-A
 - flottant programmé.
- (Le processeur AID de BOS-D n'est pas connu de BACKM).

Le processus d'alimentation en mémoire de ces deux utilitaires est le suivant :

- si la zone de mémoire centrale allouée aux processeurs du "background" (commandes BACK et JOB) est entièrement incluse dans la zone [0, 32 K - 1], alors ces utilitaires sont alimentés en fond de la zone processeur,
- sinon : la commande est refusée si l'utilitaire est le flottant (FLOAT), on a vu en effet que cet utilitaire ne peut pas être exploité au delà de 32 K.

La taille de la partition moniteur est dépendante du nombre et de la taille des options que l'utilisateur désire utiliser pendant la vacation "background". La taille minimale de la partition, c'est-à-dire sans option, doit être de 2 K mots.

- Gestion mémoire de la partition moniteur BACKM



- "Bufferisation" des entrées-sorties

OPTION BUFFER, nbuf, nsect cr

Cette commande réserve "nbuf" (buffers) de "nsect" (secteurs) qui seront alloués à chaque ouverture ou création de fichier exécutée par le système (commandes OPEN, CREATE, CALL. . .).

- "Floppy Disk Management Monitor"

FDM doit être intégré dans l'article : SFDM de la bibliothèque BIBACK - : S au moyen de l'utilitaire FUP6.

L'opération de chargement et lancement est réalisée automatiquement par la commande :

OPTION FDM cr

L'utilisation des disques souples est la même sous BACKM que sous BOS-D (voir le chapitre utilisation des disques souples du manuel de référence BOS-D).

Remarque : Dans le cas où FDM a été intégré sous RTES-D (commande EXEC), le "background" a accès au sous-programme FDM. L'option FDM n'est donc pas nécessaire pour utiliser les disques souples en "background".

Le moniteur BACKM signale l'erreur :

ERB 29

- si l'utilitaire est le module DRIP16, BACKM essaie de l'alimenter au fond de sa propre partition ; sa taille doit pour cela être égale à 3 K mots (2 K mots pour , 1 K mot pour DRIP16-A).

Faute de place suffisante, BACKM signale l'erreur :

ERB 29

6.4 - MISE EN OEUVRE DE

Si l'on dispose d'une cartouche de génération, le fichier supportant le binaire translatable du moniteur BACKM est accessible dans l'unité fonctionnelle E2 sous le nom :

BTBACK - BD

D'une manière générale, les commandes à émettre sont les suivantes :

CALL BUILD

BI BTBACK-BD, fu

MLOD adresse, [priorité], 50

CATAL IM, BACKM - : S

fu représente l'unité fonctionnelle disque supportant le fichier BTBACK-BD
adresse représente l'adresse de la partition occupée par le moniteur "background"
priorité représente le niveau de priorité logiciel associé à BACKM
Par défaut il s'agit de 126.

La FU disque support du fichier : S est quelconque. Ce n'est pas nécessairement D2.
Le fichier image mémoire ainsi obtenu est celui qui est spécifié dans la commande TASK lors de l'intégration du moniteur "background" sous RTES-D.

La bibliothèque BIBACK - : S est nécessaire si l'utilisateur emploie la commande OPTION sous BACKM. Cette bibliothèque se trouve sur la cartouche de génération :

Sous le contrôle de BOS-G, émettre les commandes :

- * CALL, FUP3
- * FDUP, BIBACK - : S, E2, BIBACK - : S, D2

6.5 - LES NUMEROS DES ERREURS DETECTEES PAR

BACKM détecte un certain nombre d'erreurs et les signale en imprimant des messages du type :

ERB n [inf1 [inf2]]

où : n représente le numéro de l'erreur.

Dans certains cas le système fournit une et parfois deux (dans le cas du défaut périphérique) informations supplémentaires.

Une erreur est spécifique de BACKM et ne peut être rencontrée avec BOS-D :

Lors de l'émission de la commande BACK débutant une vacation "background", les FU disque D1 à D8 ne sont accessibles au "background" qu'en lecture, l'autorisation d'accès en écriture ne pouvant être donnée que par une commande GIVE.
Pour les FU disque autres que D1 à D8, l'accès en lecture et en écriture est réalisé dès l'émission de la commande BACK.

Lorsqu'une demande d'écriture est adressée à IOCS et porte sur une FU disque non accessible il y a impression du message :

ERB 19 numéro de la FU

Il en est de même pour toute demande portant sur une unité fonctionnelle de numéro inférieur à 25 (unité fonctionnelle F5) à laquelle n'a été affectée par une commande GIVE aucune unité fonctionnelle de l'installation.

Toute commande nécessitant un espace mémoire (CALL, RUN, OPTION...) ne peut être émise qu'à l'intérieur d'un "job", la commande JOB comportant en paramètre la taille de la zone attribuée aux processeurs (système ou utilisateur). Hors d'un "job" aucune zone mémoire n'est affectée au "background"; l'émission d'une telle commande provoque alors l'impression du message :

ERB 27

La commande d'activation du moniteur (commande BACK) spécifie le numéro de la première partition allouable à l'activité Background.

La taille requise pour l'exécution d'un "job" détermine la zone mémoire allouable. L'un des 4 cas suivants :

- la zone mémoire allouable se situe à cheval sur 2 pages consécutives de 64 K,
 - la zone mémoire allouable déborde sur la partition du moniteur BACKM,
 - la zone mémoire allouable déborde sur la zone des tables système,
 - la zone mémoire allouable déborde sur la CDA,
- provoque l'impression du message :

ERB 34

LISTE DES ERREURS DETECTEES PAR BACKM

Numéro	Signification	Information 1	information 2
00	Alarme interne	Numéro d'alarme	
01	Instruction ACTD	Paramètre de l'ACTD	
04	Interruption non gérée par IOCS		
05	Time-out sur le périphérique de dialogue		
06	Commande inconnue ou incorrecte		
07	Affectation impossible ou incorrecte.....	Compte rendu de FMS si demande d'association SU-FICHER	
10	Appel opérateur «4 coups»		
13	Défaut périphérique	Numéro de l'unité fonctionnelle	Mot d'état de l'unité physique
14	Appel opérateur	Numéro de l'unité fonctionnelle	
16	Requête rejetée par le système pour communication d'une adresse incorrecte ou violant la protection mémoire		
17	Plus de place dans la table d'enregistrement des commandes	Nombre de tables de clés enregistrées par BACKM	
18	Requête dont le numéro n'est pas géré par BACKM	Numéro de la	
19	FU non affectée au "background"	Numéro de l'unité fonctionnelle	
20	Requête NEWS introduisant un numéro de requête interdit	Numéro de la requête	
21	Requête NEWS introduisant en "Background" le numéro d'une requête introduite par une option.	Numéro de la requête	
22	Requête interdite pour sous-dimensionnement de la zone pointée par K	Numéro de la	
23	Requête interdite en mode esclave		
24	FU non géré par IOCS	Numéro de la FU sur laquelle porte l'échange	
25	Processeur ne peut-être activé par manque de place en mémoire Processeur en mode maître buildé à une adresse incorrecte.	Adresse début zone disponible	Adresse fin zone disponible

Solar 16

Numéro	Signification	Information 1	Information 2
26	Commande interdite au cours d'un "JOB"		
27	Commande interdite à l'extérieur d'un "JOB"		
29	Option ou Utilitaire ne peut être alimenté	[Compte rendu du "builder"	
30	Commande donnant lieu à une requête rejetée par FMS	Compte rendu de l'échange (*)	
31	Fichier incorrect (ne constitue pas une image mémoire)		
32	Commande interdite en mode train de travaux		
33	Processeur utilisateur ne peut être lancé en mode maître		
34	Taille requise pour l'exécution du "job" trop importante (débordement des partitions allouables au "Background")		
35	Taille requise pour l'exécution du "job" supérieure à la taille maximum de "swapp" définie à la configuration de RTES-D		
40	Requête adressée à IOCS, transmise à FMS si association SU-fichier et rejetée par FMS	Compte rendu de l'échange (*)	
41	Requête rejetée par FMS (erreur matériel ou erreur logique de numéro supérieur ou égal à ' 20)	Compte rendu de l'échange (*)	
42	Requête de chargement de branche (BCHLR ou BACK) rejetée par FMS		
43	Erreur détectée par FDM	Compte rendu FDM (**)	
50	Commande TPIO rejetée par IOCS	Compte rendu de l'échange	
51	Commande TPIO portant sur un dérouleur non géré par IOCS		
65	OPTION déjà intégré		
72	Structure d'"overlay" interdite aux options		
83	Erreur dans l'exécution d'une requête "foreground"	Compte rendu de la requête	

(*) Se reporter à la liste des paramètres de retour fournis par FMS

(**) Se reporter au manuel de référence de BOS-C

7 - DIALOGUE INITIAL DE CONFIGURATION

Ce dialogue permet à l'utilisateur de définir le nombre effectif des partitions utilisées, le nombre maximum des tâches de son application gérées instantanément par RTES-D, la carte mémoire (zones "système", zone des données résidentes, partitions), le numéro de la tâche de l'application et le fichier de commande à activer au cas de "Restart" automatique ou de réinitialisation du système, et la configuration en nombre et en taille des systèmes sauvegardés dans la FU "Bootstrap".

Lorsque le système généré comprend les fonctions de gestion du "background", une question supplémentaire est posée concernant la taille maximum de la zone de "swap" sur disque. La réponse fournie permet de réserver, en fond d'unité fonctionnelle disque D1, l'espace nécessaire à la sauvegarde des processeurs du "background". La longueur de cette zone doit être au moins égale à la somme des longueurs des partitions utilisées par les divers processeurs du "background" et ne peut pas dépasser 64 K mots.

L'exemple ci-après est proposé à titre indicatif ; en particulier, l'encombrement du système dépend des "drivers" IOCS nécessaires à l'installation et des modules du système de fichiers FMS intégrés à RTES-D lors de la génération.

Commande soulignée : tapée au clavier du téléimprimeur de service.

*INIT, 1

→ Lancement du système sous BOS-D.

RTES-D "monte" le disque support du système. Si le montage ne peut être fait, RTES-D édite sur TS le message :

"ERREUR MONT DISQUE SYSTEME NO (a) CR = ' (b)

- a) 1 = SVC FMS "delete FU"
- 2 = SVC IOCS Read structure
- 3 = SVC IOCS DMON
- 4 = SVC IOCS MONT
- 5 = SVC IOCS Create FU

b) Compte-rendu IOCS ou FMS

RTES-D contrôle l'existence du fichier RTES-D généré. En cas d'erreur édition sur TS des messages :

- OUVERTURE FICHER (a) IMPOSSIBLE '(b)
 - (a) nom du fichier
 - (b) CR FMS
- LECTURE DESC64 IMPOSSIBLE '(b)
 - (b) CR FMS

DI (HR, CR, TK, NOMFIC-CT) ?

DI FIC-SY

→ Affectation de l'unité symbolique DI pour la configuration à une FU ou à un fichier (NOMFIC est le nom du fichier, CT le catalogue).
Le nom du fichier est quelconque mais il doit être dans D2.
En cas d'erreur lors du questionnaire RTES-D recommence au début.

DO (TS, LP, ZE, HP) ?

LP

→ Affectation de l'unité symbolique DO pour la configuration.

* DEFINITION DE LA CARTE MEMOIRE *

NB DE PARTITIONS (2,128 MAXI)? **15**
P00 P14

POUR PROGRAMMES RESIDENTS (1,15 MAXI)? **1**
P00 P00

Commande encadrée : lue sur DI affecté à : - lecteur de carte
- ou fichier.
(en l'absence de réponse : mettre - une carte vierge sur CR
- ou ligne vierge dans fichier)

La numérotation des partitions commence en zéro.

* DEFINITION DES TABLES DU SYSTEME *

ZDR (0, 32767)? **200**
'0C01F ... '0BF50

ZUEP 128*(03, 32 MAXI)? **3**
'0BF4F ... '0BE50

ZIOCB 10*(01, MAXI)? **1**
'0BE4F ... '0BE46

ZWCB 65*(01,127 MAXI)? **10**
'0BE45 ... '0BBBC

ZDF 30*(07,512 MAXI)? **20**
'0BBBB ... '0B964

ZGIN 10*(01,128 MAXI)? **60**
'0B963 ... '0B70C

Sur cet exemple, le nombre de partitions résidentes est au plus égal à 15.

But de ces tables : se reporter en fin de chapitre.

Ces tables sont constituées de pavés de 128, 10, 65, 30, 10 mots respectivement et de la ZDR.

La fourchette de réponse est donnée entre parenthèses.

En l'absence de réponse (carte vierge ou **cr**), le système prend la valeur minimale de la fourchette.

L'implantation de chaque zone est indiquée (adresses initiale et finale incluses).

Elle se fait en commençant au dessus de la zone des codes instruction et en remontant.

60 mots sont réservés en fond de mémoire pour la sauvegarde d'informations rémanentes en cas de défaut secteur.

* NOMBRE DE TACHES DE L'APPLICATION *

ZBCT 38*(01,120 MAXI)?
'0B70B...'0AE9C

* DEFINITION DES PARTITIONS *

DÉBUT DES PARTITIONS (MINIMUM '2B10)?

Pour chaque tâche, RTES-D réserve un Bloc de Contrôle de Tâche de 38 mots :

- 5 mots pour gérer l'option DAP 16 (flottant double précision)
- 12 mots de PST,
- 21 mots pour gérer les appels de tâche, l'allocation de partitions, l'"overlay", les alarmes et erreurs sur requêtes.....

En l'absence de réponse, les partitions sont implantées à l'adresse indiquée (immédiatement après le système).

Il peut être intéressant de se recadrer sur une adresse "ronde" par exemple pour l'implantation ultérieure de tâches "Buildées" en mode maître.

Si la réponse donnée n'est pas un multiple de 16 le système recadre automatiquement l'adresse sur le multiple de 16 immédiatement supérieur (cf. fonctionnement du DRPS sur la série SOLAR 16).

P00 ('10, '7E90)? '400
'03000... '033FF

P01 ('10, '7A90)? '1000
'03400... '043FF

P02 ('10, '6A90)? '1000
'04400... '053FF

P03 ('10, '5A90)? '1000
'05400... '063FF

La taille des partitions est précisée par un nombre décimal ou hexadécimal.
Une partition ne peut pas avoir une taille supérieure à (64K-1) mots et inférieure à 16 mots.

Elle est automatiquement recadrée sur une valeur égale à un multiple de 16 mots (cf. fonctionnement du DRPS).

Le système signale l'implantation en mémoire de chaque partition, et précise entre parenthèses l'espace mémoire résiduel.

Sur cet exemple, la partition l'est la première partition réservée aux programmes non résidents.


```

P04 ('10, '4A90)? [1000]
'06400 ... '073FF
*****

P05 ('10, '3A90)? [1000]
'07400 ... '083FF
*****

P06 ('10, '2A90)? [1000]
'08400 ... '093FF
*****

P07 ('10, '1A90)? [1000]
'09400 ... '0A3FF
*****

P08 ('10, '0A90)? [A90]
'0A400 ... '0AE8F
*****

P09 ('10, '8000)? [1000]
'10000 ... '10FFF
*****

P10 ('10, '7000)? [1000]
'11000 ... '11FFF
*****

P11 ('10, '6000)? [1000]
'12000 ... '12FFF
*****

P12 ('10, '5000)? [1000]
'13000 ... '13FFF
*****

P13 ('10, '4000)? [1000]
'14000 ... '14FFF
*****

P14 ('10, '3000)? [3000]
'15000 ... '17FFF
*****

```

Lorsque cette configuration est conversationnelle (DI = TK) l'opérateur dispose, en cas d'erreur, des caractères habituels de suppression de caractère (↑) ou de commande (←).

En répondant simplement N (Non !) à la place d'un nombre décimal ou hexadécimal, il peut en outre, quel que soit l'avancement de sa configuration, repartir "à zéro" sur les premières questions.

La taille maximale possible est la taille résiduelle dans la page de 64 K en cours de configuration.

NO DE TACHE A LANCER EN CAS DE "RESTART"
OU SUR INIT ? **25**

En l'absence de réponse, aucune tâche ne sera activé sur "restart" automatique ou INIT.

Si au moment du "restart" ou INIT, la tâche précisée ici n'est pas connue de RTES-D, RTES-D éditera le message : ERC 02 (tâche inexistante)

Si la tâche précisée est connue de RTES-D au moment du RESTART ou INIT elle est activée avec pour paramètre d'appel :

- 0 sur RESTART
- 1 ('FFFF) sur INIT

NOM FICHER CDE A LANCER EN CAS RESTART
OU INIT ? nomfic - ct

- . Le fichier indiqué doit être sur D2.
- . En l'absence de réponse DI sera affectée à TK.
- . Au moment du restart ou de l'INIT, RTES-D affecte DI au fichier indiqué qui doit se terminer par DI TK.
- . Si ce fichier n'existe pas DI est affecté à TK.

NO DE LA PREMIERE PARTITION DE LA CDA ? **11**

NO DE LA DERNIERE PARTITION DE LA CDA ? **14**

Ces questions ne sont posées que sur SOLAR 16-65 (sur SOLAR 16-40 les mémoires OCDA ('18) et ECDA ('19) sont initialisées avec les adresses de début et de fin de la ZDR, ceci pour permettre des entrées-sorties directes en ZDR par IOCS ou FMS).

La taille maximale de la CDA est de 64 K.

Si l'on ne répond pas à la première question (retour-chariot), la seconde n'est pas posée : dans ce cas OCDA et ECDA sont initialisées à - 1.

TAILLE MAXI DE SWAP EN KMOTS (4, 64) ? **4**

La zone de "swap" est réservée en fond de FU DI (ici 128 secteurs).

Question posée seulement si le "background" est géré.

* SAUVEGARDE DE SYSTEMES SUR DISQUE *

TAILLE DE LA FU BOOTSTRAP : 1920 SECTEURS

Le système rappelle à l'usager la taille de la FU "Bootstrap" D1 (précisée dans GENIO).

SYSTEME * AD. SECTEUR * TAILLE (SECTEURS)

01 RTES-D * 0500 * 509
02 * 1009 * (247, 0767)? **247**
03 * 1256 * (247, 0520)?

Le système RTES-D vierge est déjà stocké dans la FU "Bootstrap".

Pour les systèmes suivants, l'usager indique la taille à réserver pour la sauvegarde ultérieure par SAVE.

Pour chaque système, cette taille doit être comprise entre la taille minimale et la taille maximale de l'application et dans tous les cas inférieure à la taille restante de la FU "bootstrap".

L'absence de réponse à la question relative au système n permet d'enchaîner sur la question suivante.
La configuration de la FU "Bootstrap" se termine également après définition du système n s'il ne reste pas au moins le nombre minimal de secteurs dans la FU "Bootstrap" pour le système n+1.

OK? **Y**
ENTER TIME

TIME,25/12/76,23/59/59

* **SAVE,0,2**

* **DO TS**

* **DI TK**

Fin de configuration du Système. L'application peut être configurée en séquence à partir de commandes, lues sur le lecteur de cartes ou tapées au clavier du téléimprimeur.
Avant de lancer la première tâche, initialiser la date et l'heure.

Sauvegarde du système configuré sous le numéro 2.

Rétablissement du dialogue opérateur sur la console.

Utilisation des tables du système

Les 5 tables (ou "zones") ZUEP, ZIOCB, ZWCB, ZDF et ZGIN, constituent l'espace de travail du système.

Une telle zone est un ensemble de pavés de taille égale, consécutifs en mémoire, et gérés dynamiquement.

Z U E P : Zone d'Unités d'Enregistrements Physiques

Chaque pavé a la taille d'un secteur disque et sert de "buffer" système à FMS pour la réalisation de certaines requêtes.

En outre, le moniteur utilise un pavé de cette zone pour l'exécution de certaines commandes de l'opérateur (TASK, EXEC, SAVE....).

Le nombre minimum de pavés de la ZUEP est donc égal à 2.

L'accès à ces pavés est géré par un sémaphore d'exclusion.

Z I O C B : Zone des "Input - Output Control Blocks"

Chaque pavé a la taille d'un IOCB d'accès disque (10 mots).

Un pavé contient les informations relatives aux IOCB avec lesquels le système de fichiers FMS demande les E/S à IOCS.

L'accès aux pavés est réalisé par un second sémaphore d'exclusion.

Si . N1 est le nombre de pavés de la ZUEP et

. N2 le nombre de pavés de la ZIOCB, le degré de réentrance dans FMS est croissant avec : $\text{Inf}(N1, N2)$.

Z W C B : Zone des "Working Control Blocks"

Ces pavés sont utilisés à raison de 1 par requête adressée à FMS.

Ils lui permettent d'être réentrant pour toutes les méthodes d'accès existantes.

Ces pavés font 65 mots chacun et leur nombre optimal est le minimum entre le nombre de tâches pouvant s'adresser simultanément à FMS et deux fois le nombre de PU disques gérées par FMS.

L'accès à ces pavés est géré par un sémaphore d'exclusion.

ZDF : Zone des Descripteurs de Fichiers

Le système enregistre dans cette zone :

- les unités d'accès aux fichiers ouverts à un instant donné,
- les descripteurs de fichiers ouverts à un instant donné,
- les tables de ligature de granules des fichiers ouverts en accès direct rapide
- les ressources usager définies par la requête RESDEF.

Outre les fichiers de l'application, on prévoira un fichier ouvert par le système pour chaque tâche de l'application :

- 1 - résidente avec structure d'"overlay"
- 2 - non résidente et active à un instant donné (le nombre maximum des tâches non résidentes actives à un instant donné est égal au nombre de partitions non résidentes).

Le second point est à adapter à l'utilisation des commandes ROCK et SLOW : pour une tâche en mode "ROCK" l'unité d'accès et le descripteur de fichier sont permanents.

La saturation de cette zone est une cause d'erreur fatale : système sous-dimensionné.

ZGIN : Zone de Gestion des Informations du Noyau

L'utilisation instantanée de cette zone est la suivante :

- un pavé par tâche de l'application structurée en "overlay"
- un pavé par tâche en attente d'événement (s)
- un pavé lors de chaque exécution des requêtes sur événements (SEVENT, REVENT)
- deux pavés lors de chaque exécution de la requête sur événements (SEVDEL)
- un pavé pour chaque suspension de tâche (WAIT)
- un pavé par activation immédiate d'une tâche (RUN)
- deux pavés par activation différée ou périodique d'une tâche (START(W), TRNON (W))
- un pavé par processeur intégré sous RTES-D (commande EXEC)
- un pavé lors de chaque exécution des requêtes de segmentation BCLLR et BACK

La saturation de cette zone est également une cause d'erreur fatale.

ZBCT : Zone des Blocs de Contrôle des Tâches

Chaque pavé contient la PST d'une tâche et son extension.

On a donné ci-contre un schéma spécifiant la liste des principales informations contenues dans ce pavé.

- 5		
- 4	RÉSERVÉ	
- 3	DAP	
- 2		
- 1		N° PRIORITE
0	RA	
	//	
11	SLE	
12	NUMERO D'APPEL	NUMERO USAGER
13	NUMERO DE PARTITION OCCUPEE	TYPE ET PRIORITE
14	RÉSERVÉ TRAITEMENT ERREURS SUR REQUETES ET ALARMES	
15	ADRESSE PAVE "OVERLAY" OU '0000	
16	NOM DU	
17	FICHER	
18		
19	"PUBLIC WORD"	
20	TAILLE IMAGE MÉMOIRE	
21	INDICATEURS DIVERS	
22	NUMEROS DES PARTITIONS	
23	ACCESSIBLES Fin de liste '80	
24	NOMBRE APPELS EN COURS	NOMBRE MAXI D'APPELS
25	ADRESSE PREMIER PAVE D'APPEL	
26	ADRESSE DERNIER PAVE D'APPEL	
27		FU DISQUE CONTENANT LA TACHE
28	MODE	NUMERO USAGER PUBLIC
29		N° SPHERE
30	@ début code instruction	
31	@ implantation IM	
32	(réservé)	

REGISTRES

BCT

8 - CONFIGURATION D'UNE APPLICATION

On trouvera dans ce chapitre un exemple succinct d'utilisation de RTES-D, avec, dans l'ordre :

- Intégration de tâches et de processus
- Intégration et lancement du moniteur "background"
- Impression d'états du système
- Sauvegarde d'application sur disque
- Exploitation de tâches périodique simples.

8.1 - INTEGRATION DE TACHES ET DE PROCESSEURS

```
*BI D3
*TASK, N, TASK1-AA, 7           NT=11
*TASK, N, TASK7-AA, 5           NT=17
*TASK, N, TASK8-AA, 5, 6        NT=18
*TASK, N, TASK10-AA, 7, 8, 9    NT=20
*TASK, N, TASK11-AA, 6, 7       NT=21
*TASK, N, TASK12-AA, 6          NT=22
*TASK, N, TOVER-AA, 13, 14      NT=25
```

8.2 - INTEGRATION ET LANCEMENT DU MONITEUR "BACKGROUND"

```
*BI D2
*TASK, N, BACKM- : S, 4
*BACK, 7
*GIVE, D5, CR, LP F3, T1
```

8.3 - IMPRESSION D'ETAT DU SYSTEME

```
*SYST, 0
  NT    TASK    RST    PARTITIONS    SIZE    FICNAM-PW  FU
  011   S,011   N      PO7           00548   TASK1 -AA   15
  017   S,017   N      P05           00056   TASK7 -AA   15
  018   S,018   N      P06 P05       00056   TASK8 -AA   15
  020   S,020   N      P09 P08 P07   00075   TASK10 -AA  15
  021   S,021   N      P07 P06       00067   TASK11 -AA  15
  022   S,022   N      P06           00075   TASK12 -AA  15
  025   S,025   N      P14 P13       00183   TOVER -AA   15
```

↓
Partitions accessibles

↓
Taille en nombre de mots (décimal)

*SYST, 1

ZUEP	0FDD0	0FFCF	002
ZIOCB	0FDC6	0FDCF	001
ZWCB	0FBD2	0FDC5	002
ZDF	0F9DE	0FED1	004
ZGIN	0F786	0F9DD	004
ZBCT	0EF52	0F785	007
ZDR	0FED0	0FF9F	
P00	05000	053FF	
P01	05400	063FF	
P02	06400	073FF	
P03	07400	083FF	
P04	03400	093FF	
P05	09400	0A3FF	
P06	0A400	0B3FF	
P07	0B400	0C3FF	
P08	0C400	0D3FF	
P09	0D400	0E3FF	
P10	0E400	0EF4F	
P11	10000	107FF	
P12	10800	10FFF	
P13	11000	117FF	
P14	11800	11FFF	

→ Partition réservée au
moniteur "background"

→ Première partition des
processeurs pour cette
vacation

*SYST, 2

NAME IACCES RACCES

THERE IS NOT ANY RESOURCE IN THE SYSTEM Aucune définition de
ressource par RESDEF

ERC 05

*DEST, 8

FU	SUPATCH	ATTCH	BUSY	FAULTY	CHANNEL	STATUS
08	NOT	NOT	NOT	NOT	NOT	'0000

8.4 - Sauvegarde d'applications sur disque

*SAVE, 1, 2

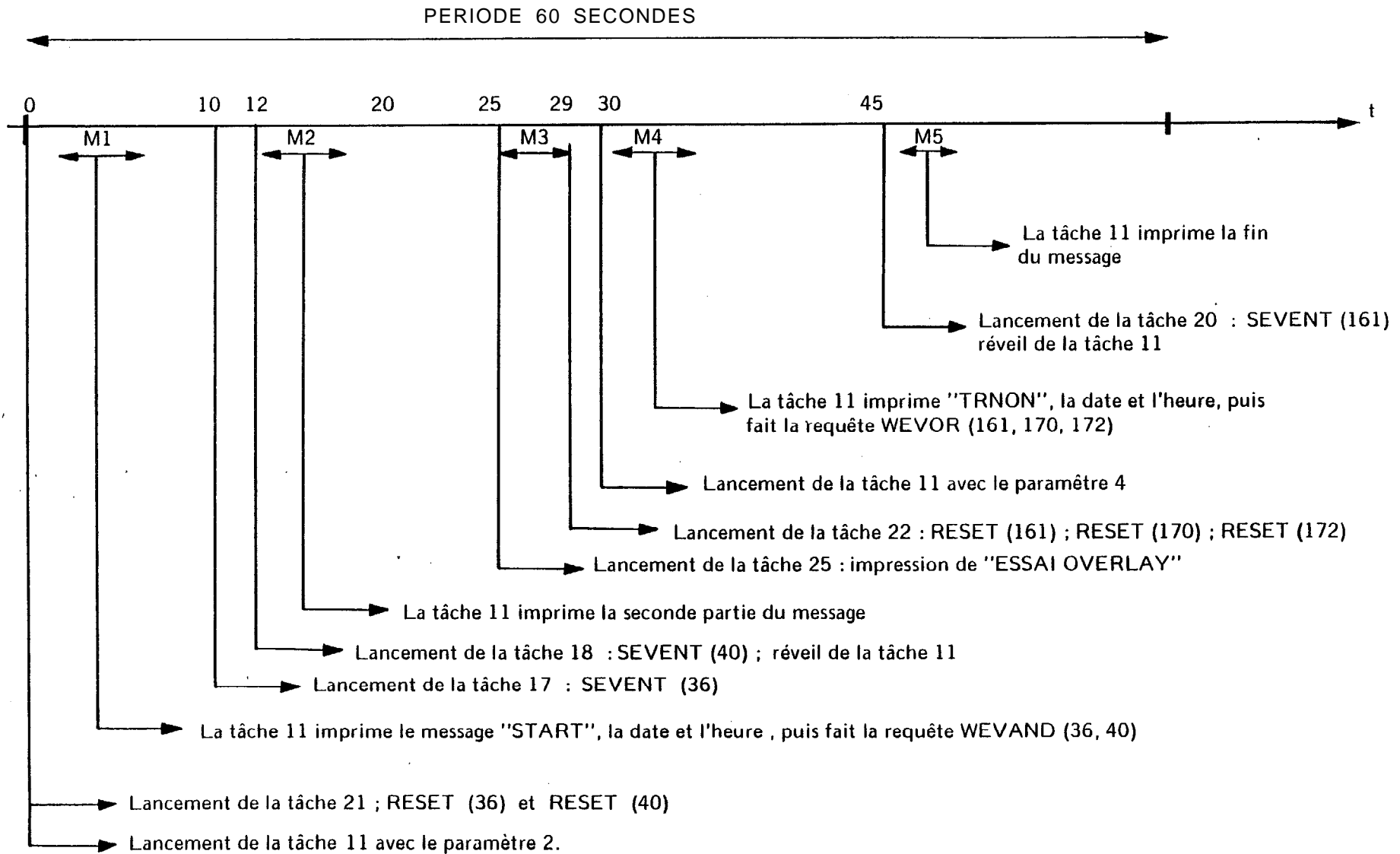
Le système 2 contient l'état actuel de la
mémoire jusqu'au fond de la partition 1

*TASK, R, TACRES-AA, 3

*SAVE, 3, 3

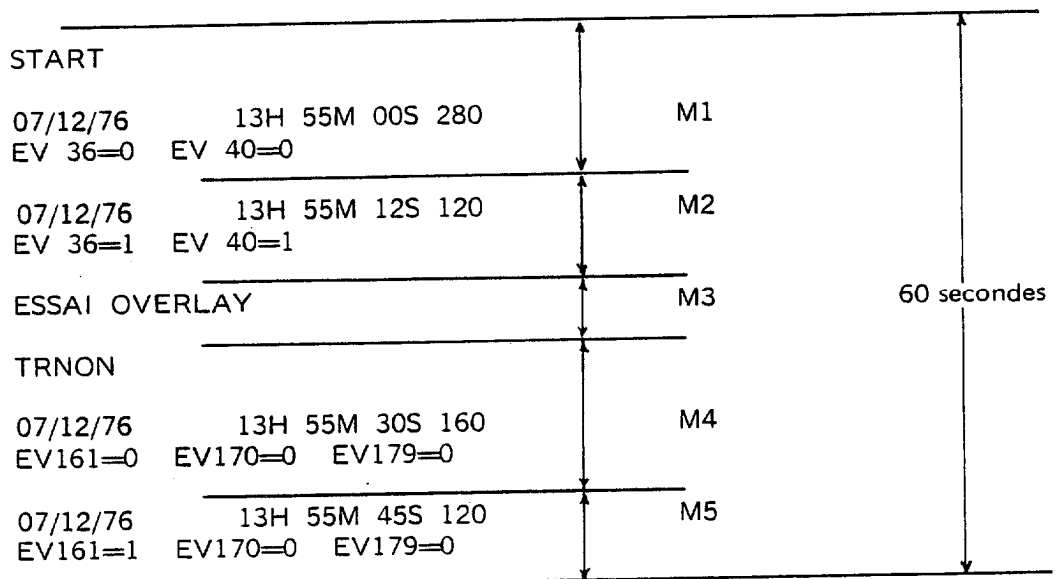
Le système 3 comprend une tâche de plus
que le précédent.

8.5 - Diagramme de temps



8.6 - Lancement de l'application

*TRUN,11,4,13/55/30,1/3	—————	Commandes d'activation du jeu de tâches
*TRUN,20,,13/55/45,1/3		
*TRUN,22,,13/55/29,1/3		
*TRUN,18,,13/55/12,1/3		
*TRUN,17,,13/55/10,1/3		
*TRUN,11,2,13/55/0,1/3		
*TRUN,21,,13/55/0,1/3		
*TRUN,25,,13/55/25.1/3		
*ELLP		
*SAF1		
*TEF2		
*TIME,7/12/76,13/54/45	—————	Initialisation de la date et de l'heure : le jeu sera lancé dans 15 secondes
*DOTS		



START

07/12/76 13H 56M 00S 200
EV 36=0 EV 40=0

07/12/76 13H 56M 12S 120
EV 36=1 EV 40=1

ESSAI OVERLAY

TRNON

07/12/76 13H 56M 30S 200
EV161=0 EV170=0 EV179=0

07/12/76 13H 56M 45S 120
EV161=1 EV170=0 EV179=0

START

07/12/76 13H 57M 00S 240
EV 36=0 EV 40=0

07/12/76 13H 57M 12S 100
EV 36=1 EV 40=1

ESSAI OVERLAY

TRNON

07/12/76 13H 57M 30S 240

9 - TRAITEMENT DU DEFAUT SECTEUR ET DU "RESTART" AUTOMATIQUE

Le SOLAR 16 est équipé d'un dispositif de détection et de traitement (câblé et microprogrammé) de la disparition et de la réapparition du secteur.

Dans RTES-D, la séquence activée au moment du "restart" automatique réalise les actions standard suivantes :

- Sauvegarde en fond de mémoire centrale des informations rémanentes* : phases d'avancement, état des événements, date et heure.
- Fermeture de tous les fichiers permanents et destruction des fichiers temporaires (EOJ généralisé).
- Chargement en mémoire de la dernière application sauvegardée par SAVE ou relancée par INIT (soit encore, chargement en mémoire de l'application en cours).
- Récupération des informations rémanentes ci-dessus.
- Activation de la tâche de l'application précisée lors du dialogue de configuration du système ; paramètre d'appel égal à 0 ; si cette tâche n'existe pas, ou si aucune tâche n'a été précisée dans ce dialogue, activation du dialogue opérateur.
- Affectation de DI au fichier de commande précisée lors du dialogue de configuration. Si ce fichier n'existe pas ou s'il n'a pas été défini DI est affecté à TK.

Dans tous les cas, il y a impression sur le dispositif associé à l'unité SA du message :

ERS 01 SYSTEM RELOAD

La tâche spécialisée de l'application activée sur "restart" dispose des informations rémanentes restituées et des fichiers sur disque dans l'état où ils se trouvaient au moment du défaut secteur.

L'heure interne devra être rétablie par dialogue opérateur dès que possible.

* La CDA (si elle existe) ne fait pas partie de ces informations rémanentes.

ANNEXE

Exemple de macro-instructions de GENIO :

Le jeu de macro-instructions ci-dessous permet de générer le moniteur d'entrées-sorties IOCS de RTES-D avec gestion de volume, correspondant à la même installation de l'exemple § 5.2.

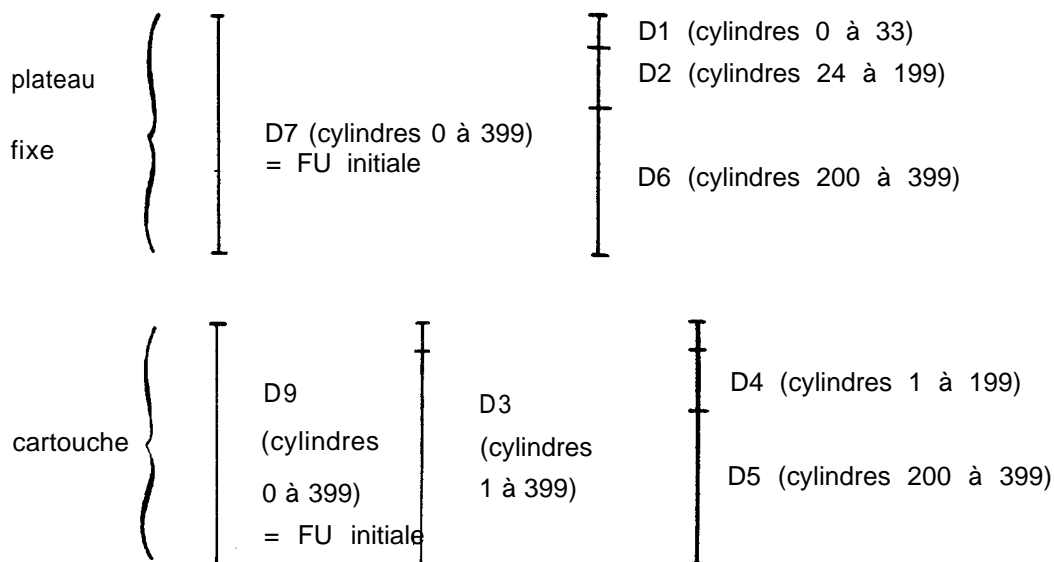
```
%RTES-D
<POUR MONTAGE VOLUME
%FSMON GVOL
%NIVEAU 13
%HTR SNIV = 0 FREQ = 50
%NIVEAU 14
%PUCD SNIV = 1 MODE = HDC ADR = '30 ITN= 1 IOP= NBV = 1
%FUICD 19 VOIE = 0 FIXE = Y
%FUESPCD 13
%FUESPCD 14
%FUESPCD 18
%FUICD 36 VOIE = 0 FIXE = N
%FUESPCD 15
%FUESPCD 16
%FUESPCD 17
%CR
%LP
%NIVEAU 15
%TTY
%HR
%ENDGEN
*END
```

Remarque : les fonctions spéciales moniteur DMONT, MONT, READ sont dans la bibliothèque BIBVOL- : S.



Exemple de macro-instructions de GENFMS :

Soit une découpe suivante en FU de l'unité physique disque:



Les macro-instructions de GENFMS à écrire pour générer les tables de FMS correspondant à cette découpe en FU et PU :

```
%PUFMS FUI = D7 OFU = 0 NBFMSMAX = 2 PUNBGRAN = 1000
%PUFMS FUI = D9 OFU = 0 NBFMSMAX = 3 PUNBGRAN = 3500
%SEQ
%IND
%DIR
%BUF
%ADR
%ALACARTE
*END
```

Distribution codes/Codes de diffusion			
Customers : Clients :			
Internal : Interne			

<p>DELIVERY ADDRESS ETIQUETTE ADRESSE</p>

Bull MTS

*1, Rue de Provence
B.P. 208
38432 ECHIROLLES CEDES / FRANCE*