

MANUEL D'UTILISATION

FONDS DOCUMENTAIRE
SMP

SOLAR

RTES1

Moniteur temps réel disque

LOGICIEL

LOGICIEL

LOGICIEL

LOGICIEL

LOGICIEL

SOLAR

RTES16
ADDENDUM A

Logiciel

SUJET : Création d'une version téléchargeable de RTES16

OBSERVATION : Première mise à jour du document 1 164 324 01 030 03
d'Avril 1985. Suivre les directives de mise à jour
et placer la présente feuille directement après la
couverture de votre manuel pour indiquer que
celui-ci est à l'indice 04.

VERSION LOGICIEL :

DATE : JANVIER 1986

(C) Bull-Sems 1986

Imprimé en France

Vos suggestions sur la forme et le fond de ce manuel seront les bienvenues. Une feuille destinée à recevoir vos remarques se trouve à la fin du présent manuel.

Ce document est fourni à titre d'information seulement. Il n'engage pas la responsabilité de Bull-Sems en cas de dommages résultant de son application. Des corrections ou modifications au contenu de ce document peuvent intervenir sans préavis ; des mises à jour ultérieures les signaleront éventuellement aux destinataires.

AVANT PROPOS

Certains chapitres de cette notice font référence au système d'entrées-sorties IOCS et au système de fichiers FMS.

La mise en oeuvre d'une application sur ce système disque trouve son point de départ dans l'utilisation du système de production de programmes BOS16.

En outre, la philosophie de RTES16 ainsi qu'un certain nombre de ses services reposent sur les fonctions «système» microprogrammées du Solar 16.

L'intérêt porté à la lecture de ce manuel s'agrémentera donc d'une assimilation aisée de son contenu si elle est supportée par la connaissance des notices suivantes :

- Manuel de Référence du Solar 16.
- Manuels de Référence et d'Utilisation d'IOCS16.
- Manuel de Référence et d'Utilisation de FMS16
- Manuel de Référence de BOS16.

Dans ce manuel, les références à IOCS16 et FMS16 sont remplacées respectivement par IOCS et FMS.

| SOMMAIRE | Page |
|---|------|
| 1 - INTRODUCTION | 1-1 |
| 1.1 - RTES16 système de 64K mots à 1024K mots | 1-1 |
| 1.2 - RTES16 et la périphérie industrielle | 1-1 |
| 1.3 - RTES16 et les langages évolués | 1-2 |
| 1.4 - RTES16 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 |
| 2.3 - Production d'un programme privilégié | 2-21 |
| 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 PL16 | 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 - Utilisation de IOCS | 4-1 |
| 4.2 - Utilisation de FMS | 4-1 |
| 4.3 - Exploitation sous les systèmes disques standards | 4-1 |
| 4.4 - Principe d'utilisation des disques à tête mobile | 4-2 |
| 4.4.1 - Initialisation du disque | 4-2 |
| 4.4.2 - Opération de montage de volume | 4-2 |
| 4.5 - Configuration en FU des disques | 4-2 |
| 5 - GENERATION D'UN SYSTEME RTES16 | 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 - GRT16 génération de RTES16 | 5-5 |
| 5.4.1 - Macro-instructions de RTES16 | 5-7 |
| 5.4.2 - Remarques | 5-7 |
| 5.4.3 - Exemple de génération de RTES16 | 5-8 |
| 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 - CONFIGURATION DU SYSTEME : CONF16 | 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 | |

1 — INTRODUCTION

RTES16 permet l'exploitation d'une installation ayant une taille mémoire comprise entre 64K mots et 1024 K 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 œuvre du système RTES16, du moniteur d'enchaînement de travaux ou moniteur "Background" intégré en option sous RTES16, et donne quelques exemples simples de réalisation de tâches de l'application en langage PL16 et FOR-TRAN temps réel.

1.1 - RTES16, SYSTEME DE 64K MOTS A 1024K MOTS

Lors de la génération du système, l'espace mémoire utilisateur est découpé en partitions dont le nombre (128 au maximum) et la taille (64K mots maximum) sont configurés aux besoins de l'application. En règle générale, les tâches intégrées dans ces partitions sont en mode esclave ou en mode privilégié : cette règle est impérative pour les tâches (résidentes en mémoire centrale ou non résidentes) susceptibles d'être exécutées au delà de 64K.

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. 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 - RTES16 ET LA PÉRIPHÉRIE INDUSTRIELLE

Les modules d'échanges (ou drivers) des périphériques industriels développés sur le SOLAR 16 peuvent être intégrés au système IOCS et utilisés sous RTES16. 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 œuvre de ces modules.



1.3 - RTES16 ET LES LANGAGES EVOLUES

Les bibliothèques BIBRTE des sous-programmes d'interface langage système permettent l'écriture de tâches à caractéristiques temps réel en FORTRAN, PL16, FORTRAN77 ou PASCAL. Ces bibliothèques donnent accès aux services spécifiques de RTES16.

- 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 - RTES16 ET L'APPLICATION

Ci-jointes quelques caractéristiques "logiciel" d'une application sous RTES16.

- 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 "bootstrapable" en mémoire centrale : 8.

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 RTES16, 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 PL16 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" et comprend :

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

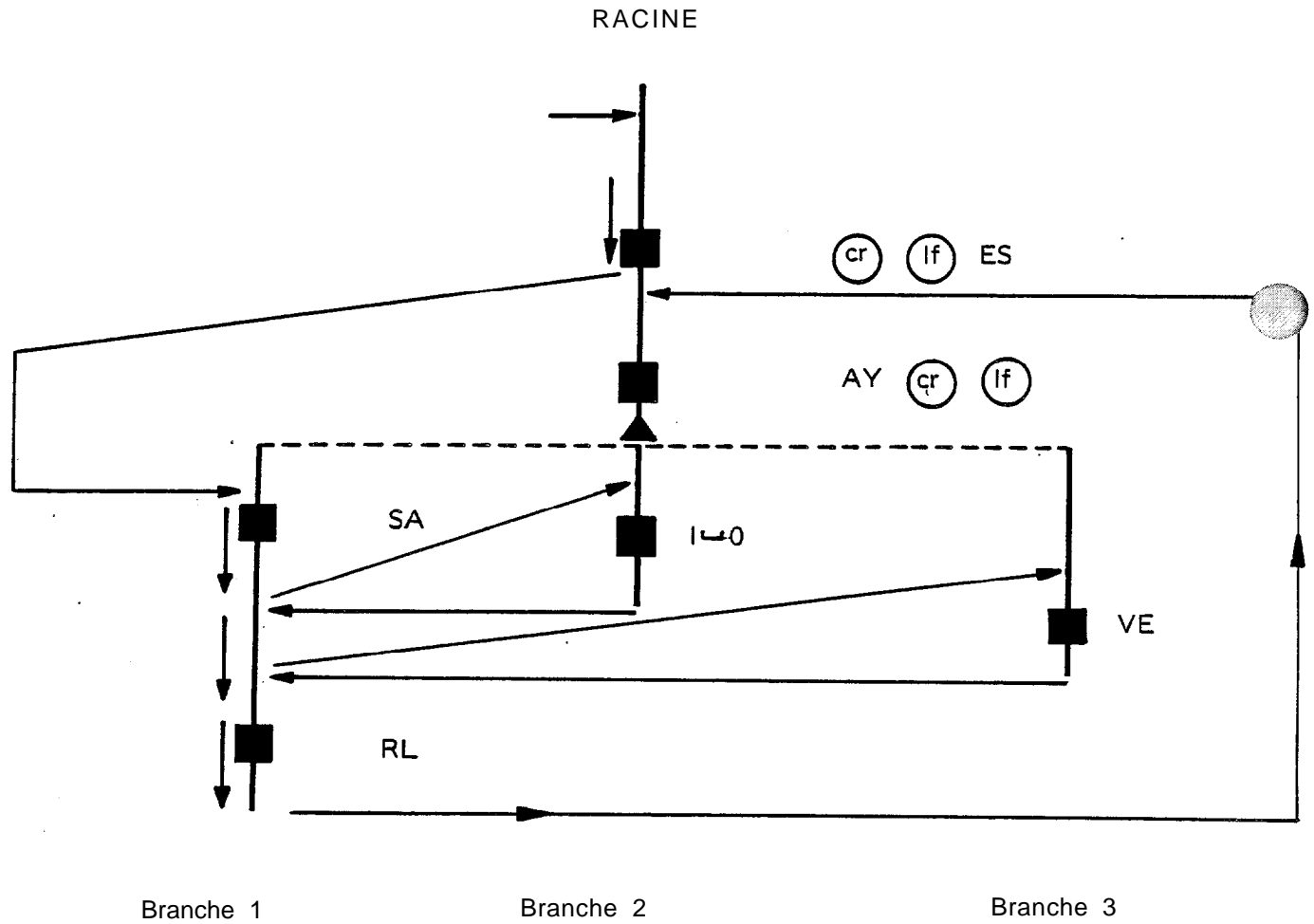
Son but global est d'imprimer le message :

(cr) (lf) ESSAI 'OVERLAY* (cr) (lf)

à chaque activation.

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

- la racine (TOVER) imprime (cr) (lf) ES , puis appelle la branche 1 (TOVER1).
- cette branche imprime SA puis appelle la branche 2 (TOVER2)
- celle-ci imprime une autre partie du message, LO puis retourne à la branche 1.
- la branche 3 (TOVER3) 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 TOV,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

```

0001 1F 1F 1F 1F 0000 0000 0000 00 00 : <<----->>
0002 1F 1F 1F 1F 0000 0000 0000 00 00 : <<          P R O C E D U R E   P 1 5          >>
0003 1F 1F 1F 1F 0000 0000 0000 00 00 : <<----->>
0004 1F 1F 1F 1F 0000 0000 0000 00 00 : <<          R A C I N E          >>
0005 1F 1F 1F 1F 0000 0000 0000 00 00 : <<----->>
0006 1F 1F 1F 1F 0000 0000 0000 00 00 :
0007 1F 1F 1F 1F 0000 0000 0000 00 00 :
0008 1F 1F 1F 02 0C00 0000 0000 01 01 :
0009 1F 1F 1F 02 0050 0000 0000 01 01 :
0010 1F 1F 1F 02 0050 0000 0000 01 01 :
0011 1F 1F 1F 02 0C50 0000 0000 01 01 :
0012 1F 1F 1F 02 0C50 0000 0000 01 01 :
0013 03 1F 1F 03 0000 0000 0000 01 01 :
0014 03 1F 1F 03 0000 0000 0000 01 01 :
0015 03 1F 1F 03 0000 0000 0000 01 01 :
0016 03 1F 1F 03 0000 0000 0000 01 01 :
0017 03 1F 1F 03 0000 0000 0000 01 01 :
0018 03 1F 1F 03 0000 0000 0000 01 01 :
0019 03 1F 1F 03 0C00 0000 0000 01 01 :
0020 03 1F 1F 03 0C00 0000 0000 01 01 :
0021 03 1F 1F 03 0001 0000 0000 01 01 :
0022 03 1F 1F 03 0002 0000 0000 01 01 :
0023 03 1F 1F 03 0003 0000 0000 01 01 :
0024 03 1F 1F 03 0C03 0000 0000 01 01 :
0025 03 1F 1F 03 0003 0000 0000 01 01 :
0026 03 1F 1F 03 0003 0000 0000 01 01 :
0027 03 1F 1F 03 0003 0000 0000 01 01 :
0028 03 1F 1F 03 0003 0000 0000 01 01 :
0029 03 1F 1F 03 0003 0000 0000 01 01 :
0030 03 1F 1F 03 0004 0002 0000 01 01 :
0031 03 1F 1F 03 0005 0002 0000 01 01 :
0032 03 1F 1F 03 0C06 0002 0000 01 01 :
0033 03 1F 1F 03 0007 0002 0000 01 01 :
0034 03 1F 1F 03 0008 0002 0000 01 01 :
0035 03 1F 1F 03 0009 0002 0000 01 01 :
0036 03 1F 1F 03 0C0A 0002 0000 01 01 :
0037 03 1F 1F 03 0008 0004 0000 01 01 :
0038 03 04 1F 04 0000 0004 0000 01 01 :
0039 03 04 1F 04 0000 0004 0000 01 01 :
0040 03 04 1F 04 0005 0004 0000 01 01 :
0041 03 04 1F 04 0C06 0004 0000 01 01 :
0042 03 04 1F 04 0009 0C04 0000 01 01 :
0043 03 04 1F 04 0009 0004 0000 01 01 :
0044 03 04 1F 04 0009 0004 0000 01 01 :
0045 03 04 1F 04 0C09 0004 0000 01 01 :
0046 03 04 1F 04 0000 0004 0000 01 01 :
0047 03 04 1F 04 0000 0004 0000 01 01 :
0048 03 04 1F 04 0000 0004 0000 01 01 :
0049 03 04 1F 04 0000 0004 0000 01 01 :
0050 03 04 1F 04 0000 0004 0000 01 01 :
0051 03 04 1F 04 0000 0004 0000 01 01 :
0052 03 04 1F 04 0000 0004 0000 01 01 :
0053 03 04 1F 04 0C00 0004 0000 01 01 :
0054 03 04 1F 04 0000 0004 0000 01 01 :
0055 03 04 1F 04 0000 0004 0000 01 01 :

<<----->>
<<          P R O C E D U R E   P 1 5          >>
<<----->>
<<          R A C I N E          >>
<<----->>
MAIN PROCEDURE TOVER
.KSTORE SECTION KTOV
RES 90;<<-----1
<<
<< DIMENSIONNER LA KSTORE EN FONCTION DES
<< BESOINS ET DES REQUETES UTILISEES
.COMMON SECTION COMTOV
INSTRUCTION SVC (3,'1C00);
EXT PROCEDURE TOVER;<<-----2
<< DECLARATION D'EXTERNE NECESSAIRE POUR
<< INITIALISATION CORRECTE DE L'ADRESSE DE
<< LANCEMENT DE LA TACHE DANS LA PST
<< CF LIBELLE 3 CI-DESSOUS
<<
WORD FCB0B1=("TD"),
      FCB1B1=("VE"),
      FCB2B1=("R1");
CONSTANT IOCS=8,
          CEXIT=23,
          BCHLR=61,
          OUT='2000,
          EMOD='9000,
          TTE='88;

ARRAY 2 WORD MESTOVE=('8DDA,"ES");
WORD IOCBTOVE0=(OUT+EMOD+TTE),
      IOCBTOVE1=(0),
      IOCBTOVE2=(4),
      IOCBTOVE3=(0),
      IOCBTOVE4=(0),
      IOCBTOVE5=(0);
ARRAY 2 WORD FINTOVE=("AY", '8DDA);
.LOCAL SECTION LOCTOV
SOFT TASK 25 (
      RC=COMTOV,
      RL=LOCTOV,
      RK=KTOV,
      START=TOVER,<<-----3
      << INITIALISATION ADRESSE LANCEMENT DE LA TACHE
      <<
      XTNUSR=25*256, << NT=25 USR=0-----4
      << INITIALISATION DES 4 MOTS EXTENSION DE PST
      << LE PROGRAMME CONSTRUIT A PARTIR DE CES 4
      << PROCEDURES COMPORTE UNE SEULE DECLARATION
      << DE TACHE;C'EST UNE TACHE EN MODE ESCLAVE,
      << DE TYPE SOFTWARE ET DE PRIORITE 25
      << RAPPEL:
      << NT (0,119)
      << USR /=128,129
      << XAPPMAX (0,255)
      << XSYSTEM RESERVE
  
```

```

LIGNE RC RL RW DONNEES TABL PRDG PF BN :+SOURCE
0056 03 04 1F 04 000D 0004 0000 01 01 :
0057 03 04 1F 04 000E 0004 0000 01 01 : XAPPMAX=10, << CUMUL MAXI D'APPELS = 10
0058 03 04 1F 04 000F 0004 0000 01 01 : XPARTITION=15, << NUMERO DE PARTITION = 15
0059 03 04 1F 04 0010 0004 0000 01 01 : XSYSTEM=0);
,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 : << CHARGES 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 : IOCBTOVE1:=RA;
0069 03 04 1F 04 0010 0004 000A 01 01 : RA:=@IOCBTOVE0; <<IMPRESSION DE (RC)+(LF)+ES
0070 03 04 1F 04 0010 0004 0008 01 01 : SVC (IOCS);
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 : IOCBTOVE1:=RA;
0076 03 04 1F 04 0010 0004 0013 01 01 : RA:=@IOCBTOVE0;
0077 03 04 1F 04 0010 0004 0014 01 01 : SVC (IOCS);
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:GUTO 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
#LO LP
#IPLC

| LIGNE | RC | RL | RH | CONNEES | 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 | : << 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=(2TOVER1); <<-----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 IOCB08=(OUT+EMOD+TTE), |
| 0021 | 1F | 02 | 1F | 02 | 0002 | 0001 | 01 | 01 | : IOCB1B1=(0), |
| 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 SAI={"SA"}; |
| 0033 | 1F | 02 | 1F | 02 | 000E | 0002 | 01 | 01 | : WORD RRL={"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:=2SAI(0); |
| 0039 | 1F | 02 | 1F | 02 | 000F | 0002 | 01 | 01 | : IOCB1B1:=RA; |
| 0040 | 1F | 02 | 1F | 02 | 000F | 0002 | 01 | 01 | : RA:=2IOCB0B1; <<IMPRESSION DE SA |
| 0041 | 1F | 02 | 1F | 02 | 000F | 0002 | 01 | 01 | : SVC (IOCS); |
| 0042 | 1F | 02 | 1F | 02 | 000F | 0002 | 01 | 01 | : RA:=2FCBOB2; |
| 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 SDRB1 ON (RA>1); |
| 0045 | 1F | 02 | 1F | 02 | 000F | 0002 | 01 | 01 | : RA:=2FCBOB3; |
| 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 SDRB1 ON (RA>1); |
| 0048 | 1F | 02 | 1F | 02 | 000F | 0002 | 01 | 01 | : IOCB1B1:=2RRL; |
| 0049 | 1F | 02 | 1F | 02 | 0010 | 0002 | 01 | 01 | : RA:=2IOCB0B1; |
| 0050 | 1F | 02 | 1F | 02 | 0010 | 0002 | 01 | 01 | : SVC (IOCS); |
| 0051 | 1F | 02 | 1F | 02 | 0010 | 0002 | 01 | 01 | : SDRB1; |
| 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  PRDG  PF  BN  :*SOURCE
0056 1F  02  1F  02  0010  0002  001A  01  01  : ERB1:
0057 1F  02  1F  02  0010  0002  001A  01  01  :      GOTQ 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
#/PLC



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 | <<----->> |
| 0003 | 1F | 1F | 1F | 1F | 0000 | 0000 | 00 | 00 | << P R O C E D U R E P 1 5 >> |
| 0004 | 1F | 1F | 1F | 1F | 0000 | 0000 | 00 | 00 | << BRANCHE 2 >> |
| 0005 | 1F | 1F | 1F | 1F | 0000 | 0000 | 00 | 00 | <<----->> |
| 0006 | 1F | 1F | 1F | 1F | 0000 | 0000 | 00 | 00 | SEGMENT PROCEDURE TOVER2 |
| 0007 | 1F | 02 | 1F | 02 | 0000 | 0000 | 01 | 01 | .LOCAL SECTION LVERB2 |
| 0008 | 1F | 02 | 1F | 02 | 0000 | 0000 | 01 | 01 | INSTRUCTION SVC (3,'1C00); |
| 0009 | 1F | 02 | 1F | 02 | 0000 | 0000 | 01 | 01 | CONSTANT IOCS=8, |
| 0010 | 1F | 02 | 1F | 02 | 0000 | 0000 | 01 | 01 | BCHLR=61, |
| 0011 | 1F | 02 | 1F | 02 | 0000 | 0000 | 01 | 01 | OUT='2000, |
| 0012 | 1F | 02 | 1F | 02 | 0000 | 0000 | 01 | 01 | EMOD='9000, |
| 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 TOVER2; |
| 0016 | 1F | 02 | 1F | 02 | 0C01 | 0001 | 01 | 01 | ARRAY 1 WORD ENAVAN=@TOVER2; |
| 0017 | 1F | 02 | 1F | 02 | 0002 | 0003 | 01 | 01 | ARRAY 2 WORD MESID=(" ", " 0"); |
| 0018 | 1F | 02 | 1F | 02 | 0003 | 0003 | 01 | 01 | WORD IOCB0B2=(OUT+EMOD+TTE), |
| 0019 | 1F | 02 | 1F | 02 | 0004 | 0003 | 01 | 01 | IOCB1B2=@MESID AND '7FFF), |
| 0020 | 1F | 02 | 1F | 02 | 0005 | 0003 | 01 | 01 | IOCB2B2=(4), |
| 0021 | 1F | 02 | 1F | 02 | 0C06 | 0003 | 01 | 01 | IOCB3B2=(0), |
| 0022 | 1F | 02 | 1F | 02 | 0007 | 0003 | 01 | 01 | IOCB4B2=(0), |
| 0023 | 1F | 02 | 1F | 02 | 0008 | 0003 | 01 | 01 | IOCB5B2=(0); |
| 0024 | 1F | 02 | 1F | 02 | 0008 | 0003 | 01 | 01 | .USING RL=LVERB2; |
| 0025 | 1F | 02 | 1F | 02 | 0008 | 0003 | 01 | 01 | RA=@IOCB0B2; |
| 0026 | 1F | 02 | 1F | 02 | 0C09 | 0003 | 01 | 01 | SVC (IOCS); |
| 0027 | 1F | 02 | 1F | 02 | 0008 | 0003 | 01 | 01 | RESTORE (RL); |
| 0028 | 1F | 02 | 1F | 02 | 0008 | 0003 | 01 | 01 | SVC (BACK); |
| 0029 | 1F | 02 | 1F | 02 | 0C08 | 0003 | 01 | 01 | ERB2: |
| 0030 | 1F | 02 | 1F | 02 | 0008 | 0003 | 01 | 01 | GOTO ERB2; |
| | | | | | | | | | END. |

<<IMPRESSION DE I O

FIN DE COMPILATION 0C00 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

| 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 | : << PROCEDURE P15 >> | |
| 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,'1000); | |
| 0008 | 1F | 02 | 1F | 02 | 0000 | 0000 | 01 | 01 | : CONSTANT IDC5=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 | : EMOD='8000, | |
| 0012 | 1F | 02 | 1F | 02 | 0000 | 0000 | 01 | 01 | : TTE='99, | |
| 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 | 0000 | 01 | 01 | : WORD VE=("VE"); |
| 0017 | 1F | 02 | 1F | 02 | 0002 | 0001 | 0000 | 01 | 01 | : WORD I0CB0B3=(OUT+EMOD+TTE), |
| 0018 | 1F | 02 | 1F | 02 | 0003 | 0001 | 0000 | 01 | 01 | : I0CB1B3=(@VE), |
| 0019 | 1F | 02 | 1F | 02 | 0004 | 0001 | 0000 | 01 | 01 | : I0CB2B3=(2), |
| 0020 | 1F | 02 | 1F | 02 | 0005 | 0001 | 0000 | 01 | 01 | : I0CB3B3=(0), |
| 0021 | 1F | 02 | 1F | 02 | 0006 | 0001 | 0000 | 01 | 01 | : I0CB4B3=(0), |
| 0022 | 1F | 02 | 1F | 02 | 0007 | 0001 | 0000 | 01 | 01 | : I0CB5B3=(0); |
| 0023 | 1F | 02 | 1F | 02 | 0008 | 0001 | 0000 | 01 | 01 | : .USING RL=LVERB3; |
| 0024 | 1F | 02 | 1F | 02 | 0008 | 0001 | 0007 | 01 | 01 | : RA=@I0CB0B3; <<IMPRESSION DE VE |
| 0025 | 1F | 02 | 1F | 02 | 0008 | 0001 | 0008 | 01 | 01 | : SVC (IDCS); |
| 0026 | 1F | 02 | 1F | 02 | 0008 | 0001 | 0009 | 01 | 01 | : RESTORE (RL); |
| 0027 | 1F | 02 | 1F | 02 | 0008 | 0001 | 000A | 01 | 01 | : SVC (BACK); |
| 0028 | 1F | 02 | 1F | 02 | 0008 | 0001 | 000B | 01 | 01 | : ERB3: |
| 0029 | 1F | 02 | 1F | 02 | 0008 | 0001 | 000C | 01 | 01 | : GOTO ERB3; |
| 0030 | 1F | 02 | 1F | 02 | 0009 | 0001 | 000C | 01 | 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 EDILE
#/BI RAC.TOVE
#/ILNA

MONITEUR BACKGROUND

**
**MODULE NO 01
**

0072 TOVER
#/81 81.TOVt
#/BLNK TOVER1

**
**MODULE NU 02
**

009A TOVER1
#/BI B2.TIVE
#/BLNK TOVER2

**
**MODULE NU 03
**

0093 TOWER2
#/B† B3.TGVE
#/BLNK TOWER3

MONITEUR BACKGROUND

*
*MODULE NO 04
*

0091 TOVER3
#/RLNK
#/ELNK

```
#/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
LUCTOV '005F
TOVER1 '0099
LVERB1 '0089
TOVER2 '0092
LVERB2 '008A
TOVER3 '0090
LVERB3 '0088

TASK S,'19
BEGIN '0000
END '00B5
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

2.3 - ECRITURE D'UN PROGRAMME PRIVILEGIE

Les programmes fonctionnant en mode privilégié sous RTES16 peuvent être des tâches ou des requêtes.

Dans les 2 cas, ces programmes seront buildés en mode esclave. C'est au moment de l'intégration au système que l'on précisera le mode de fonctionnement.

Ces programmes peuvent exécuter toutes les instructions de la machine (sauf XCTX) et peuvent accéder à toute la mémoire au moyen des instructions BLA, BSTA, BMOVE . . . (voir manuel Unités Centrales).

Par contre, l'utilisation de LAR, STAR, RD E, WOE, MVTM, MVTS est déconseillée, puisque ces instructions ne donnent que des informations relatives au registre SLO (qui est celui de la partition).

2.3.1 - Tâches

Les commandes :

TASK, NP, nomfic - ct, P1, . . Pn pour une tâche non résidente
P

et

TASK, RP, nomfic - ct, P pour une tâche résidente

permettent de préciser que la tâche fonctionne en mode privilégié.

2.3.2 - Requêtes

L'image mémoire de la requête est intégrée au système par la commande :

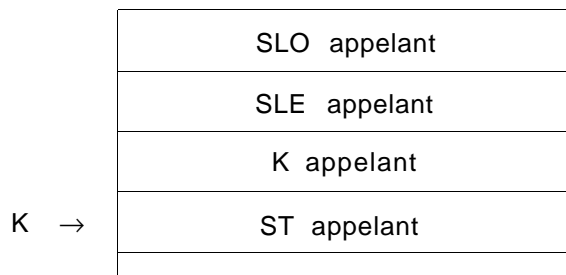
EXEC, P, nomfic - catg, P

L'adresse de la requête est communiquée au système par la SVC NEWS.

La valeur de SLO (privilégié) associé à la requête est le registre SLO lors de l'appel de la requête NEWS.

. PARAMETRES D'ENTREE :

Le registre K pointe une KSTORE située au début de la partition.
Son contenu est :



Le registre RY contient le mode de l'appelant (1 : maître, 0 : esclave).

La requête peut récupérer le SLO de l'appelant dans la KSTORE.



Le registre SLO pointe le début de la partition. Il est interdit de le modifier.

Tous les autres registres sont inchangés par rapport aux interfaces existants.

. PARAMETRES DE SORTIE :

On retourne au système par la SVC RETOUR.

On doit obligatoirement rendre la KSTORE dans le même état que ce qu'elle était à l'entrée.

. EXEMPLE :

```
PROCEDURE REQUETE
; USING RC= COM, RL = LOC ;
.
.
.
RESTORE (RC, RL) ;
SVC (RETOUR) ;
END ;

. USING RC= COM, RL = LOC, RK= KSTO ;
RA := adresse (REQUETE) ;
RY := NOREQUETE ;
SVC (NEWS) ;
SVC (ABOS) ;
END.
```

<<ATTENTION RK = KSTO
INTERDIT

2.3.3 - Le configurateur de KSTORES

Tout programme fonctionnant en mode privilégié doit posséder une kstore à l'intérieur de sa partition. Dans le cas des tâches, cela ne pose aucun problème, car elles en possèdent déjà une. Dans le cas des requêtes, puisqu'on ne peut plus utiliser la kstore de l'appelant, il faut en allouer une dans la page de la requête avant de l'appeler, et il en faut autant que de tâches pouvant accéder simultanément à la requête.

A cet usage, il est livré un configurateur de kstores qui permet de générer les tables nécessaires.

MODE D'EMPLOI :

Sous BOSD, frapper :

```
U1 KMAC-SY
CC GENK-SY
EOJ
```

Macro-instructions (fichier KMAC-SY) :

```
%KSTORE TAILLE = xxx NOMBRE = yyy
*END
```

TAILLE : nombre décimal - doit être au moins égal au nombre de mots utilisés par la requête + 5.

NOMBRE : nombre de kstores utilisables simultanément. En cas de saturation, une erreur "système sous-dimensionné" (7) est rendue à l'appelant.

Le fichier KSTORE-BO généré contient les buffers de kstore, et il est à link-éditer en tête du programme.

Les kstores doivent obligatoirement se trouver au début de la partition.

Exemple de fichier de commandes :

```
CALL PL
SI REQUET-SY
BO REQUET-BO
IPLC
```

Compilation de la requête utilisateur

```
CLOSE BO
U1 CC
CC GENK-SY
%KSTORE TAILLE = 80 NOMBRE = 40
*END
CALL LKLOAD
LINK KSTORE-BO
LINK REQUET-BO
ENDL
CATA IM, REQUET-IM
EOJ
```

3 - BIBLIOTHEQUES TEMPS REEL

3.1 - INTRODUCTION

Les sous-programmes des bibliothèques BIBRTE ont pour but d'établir l'interface entre un programme utilisateur écrit en langage FORTRAN, PL16 FORTRAN77 ou PASCAL et les requêtes programmées du système RTES16.

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 RTES16, 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) ;
```

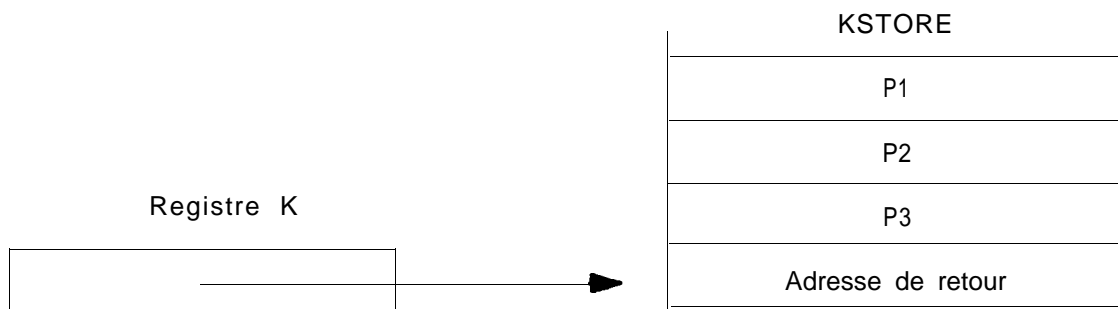



3.3 - PASSAGE DES PARAMETRES

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 RTES16, à l'exception de :

- IOCS
- F M S
- 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 RTESDi, $1 \leq i \leq 7$.

(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 RTES16 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 DES BIBLIOTHEQUES

3.9.1 - Mise en œuvre

Les bibliothèques de sous-programmes sont dans 3 fichiers indexés de nom :

- BIBRTE-RT pour FORTRAN et PL16
- BIBRTE-F7 pour FORTRAN77
- BIBRTE-PA pour PASCAL.

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 | RTESD1 | RTESD2 | RTESD3 | RTESD4 | RTESD5 | RTESD6 | RTESD7 |
|------------|-------------------|-----------------|--------|--------|--------|--------|--------|--------|--------|
| 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.

| MNÉMONIQUE | 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 | | | |

| MNÉMONIQUE | TAILLE en Mots | Prof. KSTORE | RTESD1 | RTESD2 | RTESD3 | RTESD4 | RTESD5 | RTESD6 | RTESD7 |
|------------|-------------------|-----------------|--------|--------|--------|--------|--------|--------|--------|
| FBASIC | * 38 36 | 60 | | | | | | | |
| DCLOCK | * 80 69 | 65 | | | | | | | |
| FCLOCK | * 71 62 | 65 | | | | | | | |
| CLOCK | * 186 157 | 70 | | | | | | | |
| CLOCKA | 126 | 70 | | | | | | | |
| CLOCKD | 75 | 70 | | | | | | | |
| DTIMER | * 80 69 | 65 | | | | | | | |
| FTIMER | * 71 62 | 65 | | | | | | | |
| TIMER | * 127 103 | 70 | | | | | | | |
| TIMERA | 85 | 70 | | | | | | | |
| TIMERB | 62 | 70 | | | | | | | |
| RTIMER | * 85 74 | 70 | | | | | | | |
| DETOR | * 80 69 | 65 | | | | | | | |
| FETOR | * 71 62 | 65 | | | | | | | |
| RELAY | * 109 89 | 65 | | | | | | | |
| RELAYA | 68 | 65 | | | | | | | |
| RELAYR | 38 | 65 | | | | | | | |

* Taille du sous-programme de BIBRTE-F7.

Le nombre de la ligne suivante est la taille du sous-programme de BIBRTE-RT (et du sous-programme correspondant de BIBRTE-PA lorsqu'il existe).

Remarques :

- 1) Les sous-programmes FBASIC, DCLOCK, FCLOCK, DTIMER, FTIMER, RTIMER, DETOR, FETOR ont pour nom respectivement : XFBASI, XDCLOC, XFCLOC, XDTIME, XFTIME, TIMERC, XDETOR, XFETOR dans BIBRTE-PA.
- 2) Les sous-programmes CLOCK, TIMER et RELAY n'existent pas dans BIBRTE-PA et sont remplacés chacun par deux sous-programmes spécifiques à BIBRTE-PA : CLOCKA, CLOCKD, TIMERA, TIMERB, RELAYA, RELAYR.



| 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 link-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 - UTILISATION DE IOCS

IOCS assure la gestion des FU 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 FU
- 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.

4.2 - 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).

4.3 - EXPLOITATION SOUS LES SYSTEMES DISQUES STANDARD

Les systèmes disques standards 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.



4.4 - PRINCIPE D'UTILISATION DES DISQUES A TETE MOBILE

4.4.1 - Initialisation du disque

Tous les supports disque 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.

La description complète des secteurs 3 et 4 est faite dans le manuel IOCS16.

4.4.2 - 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.

4.5 - CONFIGURATION EN FU DES DISQUES

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 et 4 qui ne sont pas protégés.

5 - GENERATION D'UN SYSTEME RTES16

5.1 - PRINCIPE

- Elle s'opère sous les systèmes BOS-G ou BOS16 grâce au processeur GRT-16.
- La génération d'un système RTES16 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 RTES16.
 - . Le support de génération contient les M.O.L. de RTES16 sous la forme d'un fichier indexé "RTES16-MO".
 - . La génération s'opère de la façon suivante :

5.2 - GENIO CONFIGURATION DE IOCS

Le moniteur d'entrées-sorties IOCS de RTES16 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 :

```
%RTES16
%NIVEAU 13
%HTR MFI
%NIVEAU 14
%PUVM SNIV = 1 MODE = HDC ADR = '30 ITN = 1 IOP = NBV = 1
%FUIVM 19 VOIE = 0 FIXE =Y
%FUESPVM 13
%FUESPVM 14
%FUESPVM 18
%FUIVM 36 VOIE = 0 FIXE = N
%FUESPVM 15
%FUESPVM 16
%FUESPVM 17
%CR
%LP
%NIVEAU 15
%MFI ASY STD
%FU 2 CDE = '0 SENS = 0 FINDIC = 0
%FU 3 CDE = '0 SENS = 1 FINDIC = 1
%HR
%HP
%ENDGEN
*EOT
```

Remarques : Plusieurs contraintes doivent être respectées pour la configuration de IOCS de RTES16.

- . La 1ère macro doit être %RTES16.
- . La gestion de volume est obligatoire sur RTES16, elle est intégrée à la macro RTES16.
- . 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 FUIVM précédente).
- . Dans le cas où l'on dispose d'un périphérique de dialogue réservé au "background" (téléimprimeur ou console de visualisation alphanumérique), les FU F7 (27) en sortie et F8 (28) en entrée sur ce périphérique seront utilisées par défaut (voir commande GIVE).
- . La macro-instruction HTR ne doit pas être omise. Elle permet de configurer l'horloge temps réel obligatoire dans la configuration matériel de RTES16. 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" BDRV16 : S.
- . Les fonctions spéciales de montage de volume sont dans la bibliothèque BVOL16 - : S.

5.3 - CONFIGURATION DE FMS

5.3.1 - Macro-instructions décrivant le moniteur de gestion de fichiers (cf. manuel d'utilisation de FMS16)

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 \leq taille des granules \leq 256 secteurs |
| | 2 \leq nombre de granules \leq 2000 |
| organisation grand disque | 3 secteurs \leq taille des granules \leq 32 K - 1 secteurs |
| | 1 \leq nombre de granules \leq 32 656 |

nombre de granules x taille des granules \leq 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 \leq | numéro d'espace \leq 15 |
| 0 \leq | adresse en cylindre du début FU < longueur du disque |
| | longueur en cylindre de la FU telle que 0 \leq longueur + adresse < longueur du disque |
| | 0 \leq longueur table d'allocation des granules \leq 125 |

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



Exemple : Opération à faire sous BOS16

Définition de la structure d'une cartouche

```
CALL FUP4
DMONT D9
SDEF, D9, LABEL, 1, 1, 339
SDEF, D9, LABEL, 2, 1, 300
SDEF, D9, LABEL, 3, 301, 399
```

Remarque :

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.

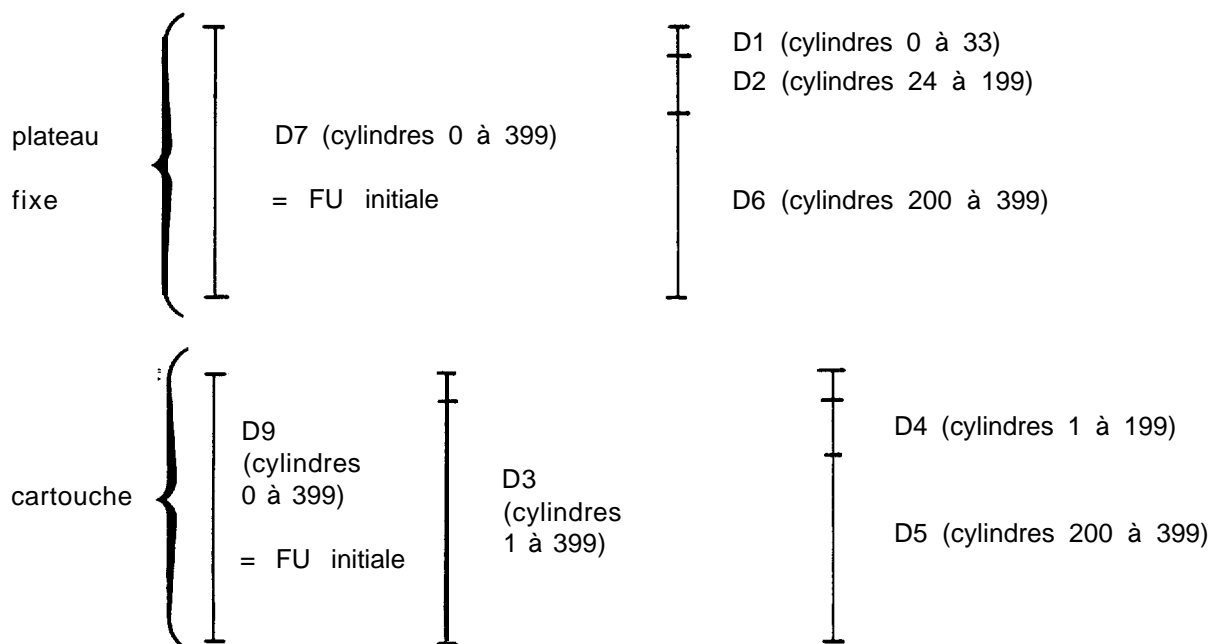
Exemple :

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



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 FMS16 correspondant à cette découpe en FU et PU :

```
%USER NB = 12
%DISK NB = 2
%ZDF NN = 100
%PUFMS FUI = D7 OFU = 0 NBFMSMAX = 2 PUNBGRAN = 1000
%PUFMS FUI = D9 OFU = 0 NBFMSMAX= 3 PUNBGRAN = 3500
%SEQ
%IND
%DIR
%ALACARTE
*END
```

5.4 - GRT16 GENERATION D'UN SYSTEME RTES16

GRT16 permet de générer l'ensemble des commandes nécessaires à la génération de RTES16. La description des différentes phases de la génération est faite à l'aide de macro-instructions de GRT16.

Les différentes phases de la génération d'un système RTES16 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 RTES16),
- création de l'image mémoire du système,
- création de la carte mémoire du système,
- implantation dans la FU D1 (commandée par la commande GSYS).



Différents types de systèmes :

Le système RTES16 peut prendre plusieurs formes :

- RTES16 classique : le dialogue est en overlay, le système gère des tâches résidentes et non résidentes.
- Option "branches résidentes" : les branches ENTREE, ANALYS, MOPASS, qui sont appelées lors de l'exécution de chaque commande, sont résidentes en mémoire centrale, ainsi que les commandes IRUN et SUFU.
Cette option permet d'économiser des accès disque.
- Système RESIDENT : il est entièrement résident en mémoire centrale.

Toutes les tâches sont du type résident. Donc la commande TASK ne sera acceptée qu'avec le paramètre "R".

Le dialogue opérateur est réduit aux commandes :

CONT, DBCT, DEST, DMON, DRUN, EXEC, INIT, IRUN, KEND, KILL, MEMO, MONT, PAGE, PRIN, PRTY, RSUM, SAVE, STAT, STOP, SUFU, SYST, TACT, TASK, TIME, TRUN, VALI

Il ne peut pas supporter le moniteur background.

- Système TELECHARGEABLE :
 - . Ce système est destiné à être téléchargé sur un frontal par une ligne asynchrone, avec utilisation de la fonction TEL-TIP du coupleur multi-fonctions.
 - . Il est en général chargé de faire de l'acquisition de données sur des lignes, de traiter ces données pour les transmettre au central.
 - . Il est de type résident et possède un restart sans rechargement du système.
 - . Le dialogue est réduit aux commandes :
CONT, DBCT, DEST, DMON, DRUN, KEND, IRUN, MEMO, MONT, PAGE, PRIN, PRTY, RSUM, STAT, STOP, SUFU, SYST, TIME, TRUN, VALI.
 - . Il peut ne pas posséder de console de service (FU TK et TS n'existent pas).
 - . Les tâches sont intégrées avant le téléchargement (par CONF16).
 - . Il peut ne pas avoir de disque ; dans ce cas, il n'y a pas de FMS, et les commandes MONT et DMON n'existent pas.
 - . La description complète du système téléchargeable et du téléchargement est faite dans le manuel TELEC16.
- Pour chacun de ces systèmes, il existe les options suivantes :
 - . Gestion du mode privilégié :
Elle n'est accessible que sur SOLAR 16-70 et 16-90.
Elle permet d'avoir FMS en mode privilégié (implanté à partir de l'adresse 64K).
Le système peut gérer des tâches et des requêtes utilisateur en mode privilégié.



. Restart sans rechargement du système :
Voir chapitre 9

. Gestion de la date et de l'heure par le coupleur multi-fonctions :

Si on précise dans les macro-instructions de GENIO la FU liée à la gestion de la date et de l'heure par le coupleur (macro-instruction % PU TRL présente), le système :

- met à jour le coupleur avec la date et l'heure rentrées lors de la commande TIME.
- lit l'heure sur le coupleur et met à jour sa date et son heure au lancement, sur STOP-INIT-RUN ou sur RESTART.

Lors d'un RESTART, le système édite sur la SU DO la durée de la coupure.

- lit l'heure sur le coupleur à tous les TOP d'horloge, ce qui permet d'être toujours à l'heure, même après un arrêt momentané du calculateur.

5.4.1 - Macro-instructions de RTES16

%FUGENE FU

Précise la FU disque qui contient les binaires nécessaires à la génération :

RTES16-MO, IOCS16-SY, GIO16-SY, FMS16- : S, GFMS16-CC, BVOL16- : S, BDRV16 - : S.

Cette macro-instruction est optionnelle. Par défaut, les bibliothèques devront se trouver sur la FU du job.

%FUGENE RTES16 = FU1 AUTRES = FU2

Cette macro-instruction peut remplacer la forme précédente. Elle permet d'avoir la bibliothèque RTES16 sur une FU (FU1) et toutes les autres sur une autre FU (FU2).

%MODPRI

Si elle est présente, cette macro permet de générer un système :

- qui gère le mode privilégié,
- qui a un FMS en mode privilégié.

Elle ne doit être utilisée que pour générer un système pour un SOLAR 16-70.

%BRANCHES RESIDENTES

Permet d'avoir les branches ENTREE, ANALYS, MOPASS, IRUN, SUFU en mémoire centrale. Ceci permet d'améliorer les performances du dialogue.

%RESTART MEMOIRE

Si cette macro-instruction est présente, elle permet de ne pas recharger le système après une coupure secteur.

Par défaut, la réapparition du secteur s'accompagnera du rechargement du système et de l'application.

%RESIDENT

Permet de générer un système entièrement résident en mémoire centrale. Le dialogue système est résident, les tâches de l'application sont toutes résidentes en mémoire.

%TELECHARGEABLE { AVEC }
{ SANS } DISQUE

Si elle est présente, cette macro-instruction permet de générer un système téléchargeable. Elle impose en outre :

- un restart sans rechargement du système (% RESTART MEMOIRE)
- un système résident (% RESIDENT)
- un dialogue réduit,
- pas de sauvegarde.

Le paramètre { AVEC }
{ SANS } permet d'avoir un FMS ou non ; si non il n'y a pas non plus de fonction de montage de volumes.

%MACRO IOCS ON { nomfic - catg [, FU] }
{ CR } [FMS ON { nomfic - ct [FU] }
{ HR }]

Précise les supports des macro-instructions d'IOCS et FMS.

%FMSG

Permet l'utilisation du noyau grand disque de FMS.

Dans le cas où la macro-instruction %MODPRI est présente, %FMSG est inutile.

En effet, dans ce cas, on génère automatiquement un FMS utilisant le noyau grand disque.

%BUILD { nomsys } { AVEC }
{ * } { SANS } BACKGROUND [FIN RACINE { HHHH }
{ * }]

Création de l'image mémoire du système.

nomsys : nom du fichier image mémoire du système. Par défaut, le nom est RTES16

Le catalogue est : S.

Dans le cas d'un système avec FMS déporté, le nom du fichier FMS est nomsys - : :

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

Edition de liens des drivers nécessaires se trouvant dans une bibliothèque de drivers.

On peut faire apparaître cette macro-instruction plusieurs fois.

Les bibliothèques BDRV16 - : S et BVOL16 - : S sont scrutées de façon automatique. Il n'est donc pas nécessaire de les faire apparaître dans les macro-instructions.

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

Edition de liens d'un module link-éditable quelconque (driver spécifique par exemple).



%FIN

Fin de la génération du système. Cette macro-instruction est obligatoire.

%SYSTEM $\left\{ \begin{smallmatrix} 0 \\ 7 \end{smallmatrix} \right\}$ NUMSECT Adresse

Provoque l'intégration du système généré dans la FU D1. Cette macro-instruction est facultative.

5.4.2 - Exemple de génération de RTES16

Macro-instructions de GRT16

%FUGENE D4

%MODPRI

%BRANCHES RESIDENTES

%MACRO IOCS ON CR FMS ON MACFMS- RT, D3

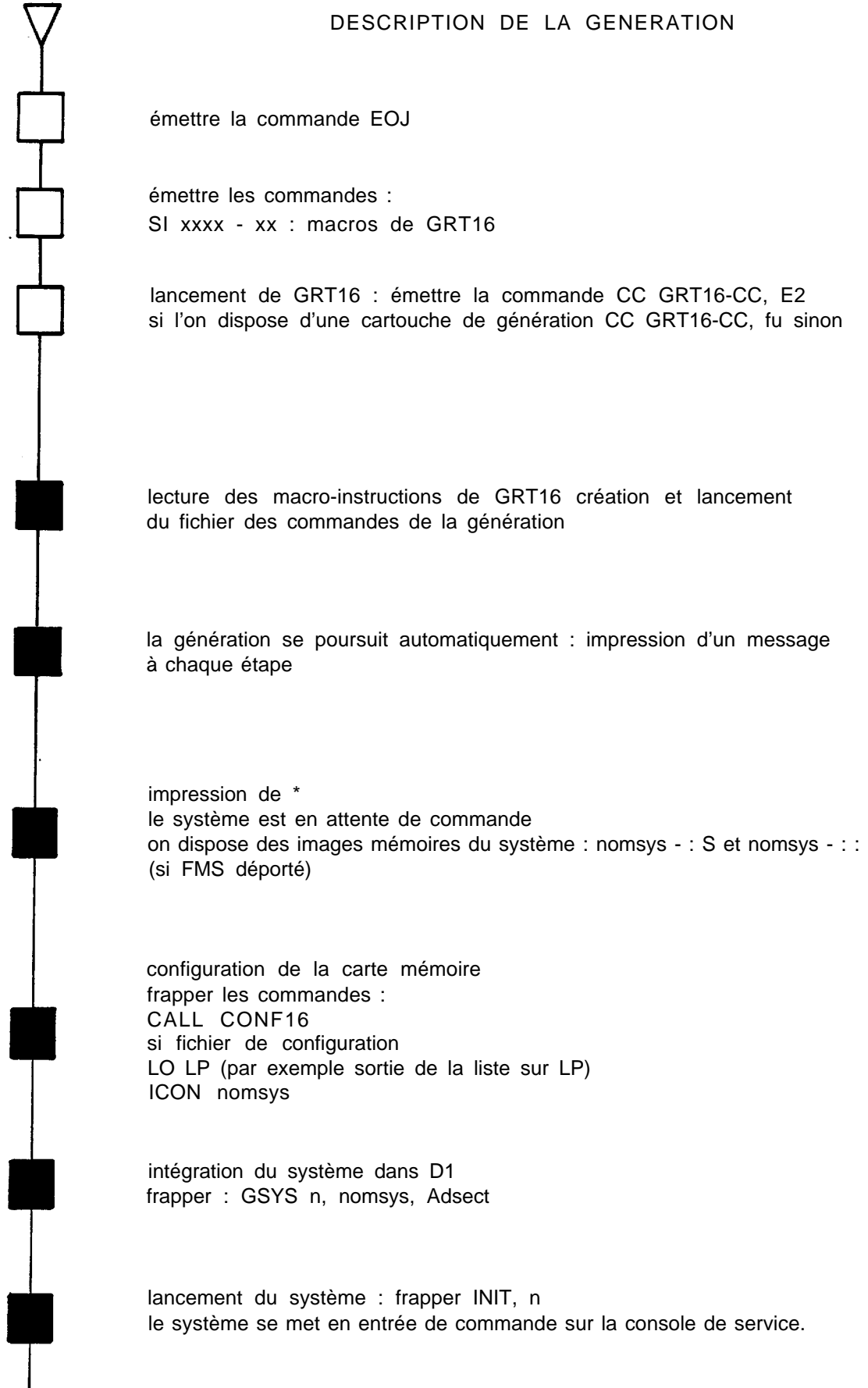
%BUILD RTES16 AVEC BACKGROUND

%BIBDRV ON DRVUTI - : S, D5

%FIN

*END

DESCRIPTION DE LA GENERATION





Intégration des tâches de l'application

Sauvegarde du système et de l'application par la commande SAVE.

En cas d'erreur pendant la génération, un message donne le diagnostic.
Les messages ont une forme différente suivant la phase de génération pendant laquelle ils apparaissent.
Pour leur compréhension se reporter au manuel de référence de BOS16, paragraphe 7.4.3.

6 - UTILISATION DU MONITEUR "BACKGROUND"

Le moniteur "background" BACKM est une tâche de RTES16 dont le but est d'apporter à l'utilisateur les services rendus par le système BOS16.

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 BOS16. Seuls sont repris dans ce chapitre les points présentant une différence entre les deux moniteurs BOS16 et BACKM.

6.1 - LES REQUETES DE BACKM

Toutes les requêtes programmées du système BOS16 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 RTES16, 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 BOS16.

Il est possible cependant de rendre toutes les requêtes spécifiques du temps réel accessibles au "background" (voir Manuel de référence de RTES16, 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 :

(cr) (lf) . (#)

D'une manière générale les commandes que peut alors émettre l'utilisateur sont celles de BOS16. 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, INIT, IBOS.

Le caractère Backspace est équivalent à la flèche ↑ (ou ^) pour annuler le caractère précédent. DEL (ASCII 'FF), puis  réaffiche la dernière commande rentrée. L'opérateur peut éventuellement la modifier à l'aide des caractères d'annulation avant de la réexécuter en frappant .

La forme des commandes reconnues par BACKM est décrite dans le manuel de référence de BOS16. Deux commandes diffèrent quelque peu :

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

| | | | | |
|-----|------------|--|-----------|---|
| JOB | nom, catg, | $\left. \begin{matrix} \{SU\} \\ \{FU\} \end{matrix} \right\}$ | ,, taille |  |
|-----|------------|--|-----------|---|

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 BIBA16. Cette bibliothèque contient en standard 2 modules : la "bufferisation" des entrées-sorties (BUFFER) et les macro-commandes (MACRO). Cependant les modules suivants peuvent être intégrés à BIBA16 :

- 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 et entiers double précision

DFLOAT doit être intégré dans l'article : SDFLOAT de la bibliothèque BIBA16 - : S au moyen de l'utilitaire FUP6.

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

OPTION DFLOAT (cr)

- Outil de mise au point DRIP16

DRIP16 doit être intégré dans l'article : SDRIP16 de la bibliothèque BIBA16 - : S au moyen de l'utilitaire FUP6.

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

OPTION DRIP16 (cr)

- Les macro-commandes :

OPTION MACRO, tbuf (cr)

Cette commande permet d'intégrer MACSP à BACKM pour assimiler les macro-commandes (voir MR de BOS16) tbuf est la taille en mots du buffer allouée pour ce faire (minimum = 2000).

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 BOS16 à RTES16 et vice versa.

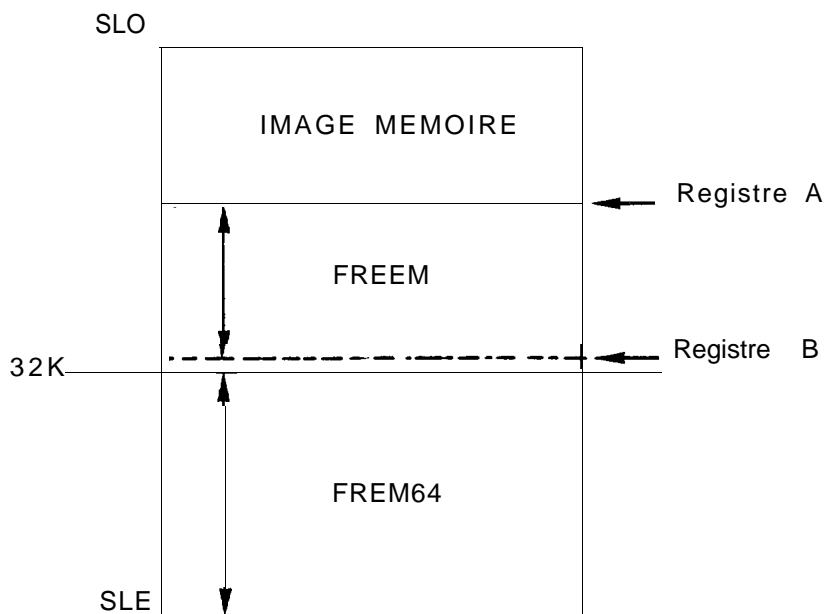
De manière à minimiser l'encombrement sur disque un même processeur doit donc pouvoir s'exécuter à des implantations mémoire différentes. Ceci n'est possible que si le processeur se déroule en mode esclave (intégration au moyen de la commande SLOD du "builder"). Le processeur est alimenté en mémoire 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.

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.



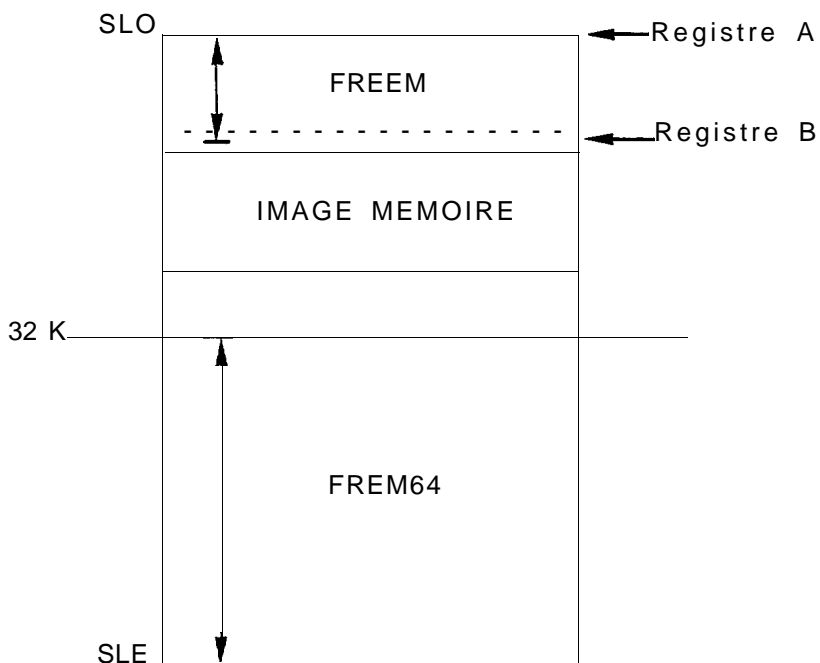
Exemple 1 - Cas adresse d'implantation égale à zéro



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'adressed'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 BOS16, les deux commandes RUN et LOAD permettent la mise en oeuvre des processeurs utilitaires suivants :

- module DRIP16-A
- flottant programmé.

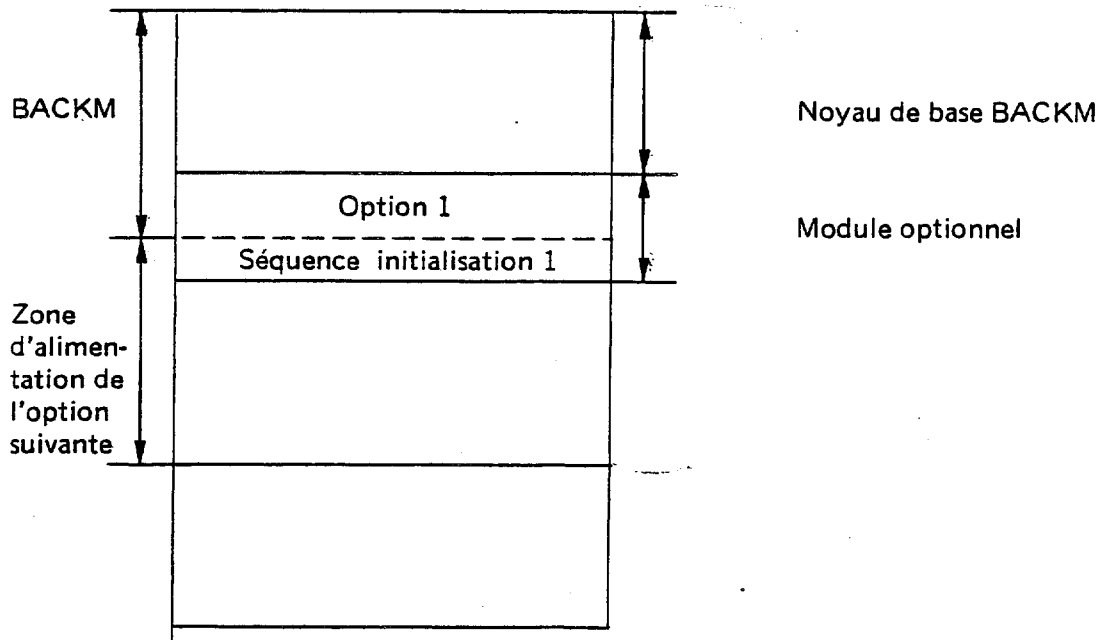
Le processus d'alimentation en mémoire de ces deux utilitaires est le suivant :

- 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 BIBA16 - : 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 BOS16 (voir le chapitre utilisation des disques souples du manuel de référence BOS16).

Remarque : Dans le cas où FDM a été intégré sous RTES16 (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 BACKM, 1 K mot pour DRIP16-A).

Faute de place suffisante, BACKM signale l'erreur :

ERB 29

6.4 - MISE EN OEUVRE DE BACKM

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 :

BACK16 - BD

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

CALL BUILD

BI BACK16-BD, fu

MLOD adresse, [priorité], 50, 64

CATAL IM, BACKM - : S

- fu représente l'unité fonctionnelle disque supportant le fichier BACK16-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 BACKM - : 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 RTES16.



La bibliothèque BIBA16 - : 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, BIBA16 - : S, E2, BIBA16 - : S, D2

6.5 - LES NUMEROS DES ERREURS DETECTEES PAR BACKM

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 BOS16 :

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 BACKM (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 |

| * 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 RTES16 | | |
| 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 - CONFIGURATION DU SYSTEME : CONF16

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 RTES16, 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 en cas de "Restart" automatique ou de réinitialisation du système.

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 RTES16 lors de la génération.

Le processeur CONF16 peut s'exécuter sous n'importe quel moniteur : BOS, TSM, TSF, BATCH de MPES ou BACKM.

- La lecture des réponses aux questions se fait sur SI.
- Une copie des réponses est écrite sur SO. Ceci permet, en affectant SO à un fichier, de créer la configuration du système au clavier de la console de service en conversationnel et de pouvoir réutiliser ultérieurement le fichier créé.
- La liste est sortie sur LO.
- Exemple :

```
*CALL CONF16
*SI TK
*SO INIT - RT
*LO LP
*ICON RTESN3          lancement de la configuration.
```

Le temps d'initialisation du processeur pouvant être assez long, il imprime une série de "." sur la SU EL.

Le fichier système (RTESN3 - : S ci-dessus), ainsi que le fichier FMS (RTESN3 - : :) s'il s'agit d'un système avec FMS déporté doivent se trouver sur la FU du job.

```
* GSYS n, RTESN3,800          intégration du système dans D1.
```

- En cas d'erreur sur une réponse, on distingue 2 cas :
 - . L'entrée des réponses se fait sur le clavier : CONF16 repose la question jusqu'à ce que la réponse soit satisfaisante.
 - . L'entrée des réponses se fait sur un fichier disque :
CONF16 émet un message et rend le contrôle au moniteur. Dans ce cas, rien n'est réécrit dans le fichier système.
- Dans le cas d'un système TELECHARGEABLE, CONF16 permet d'intégrer les tâches dans l'image mémoire du système, de créer les fichiers qui seront téléchargés. On dispose des commandes :

```
TASK [ R/P ] , nomfic - ct, P      :          intégration d'une tâche
EXEC, [ R/P ] , nomfic - ct, P      :          intégration d'un processeur
FCON                               :          fin de configuration
```

Pour plus de détails, consulter la notice TELEC16.

* MEMORY MAP *

Réponses : lues sur SI

NUMBER OF PARTITIONS (2,128 MAX)? 10
P00.....P09

(en l'absence de réponse : mettre - une carte vierge sur CR
- ou ligne vierge dans fichier)

La numérotation des partitions commence en zéro.

FOR RESIDENT PROGRAMS (1,10 MAX)? 2
P00.....P01

Les programmes résidents sont :
- les tâches résidentes de l'application, intégrées par la commande TASK, et les processeurs intégrés par la commande EXEC
- cette question n'apparaît pas si le système est du type résident ou téléchargeable.

* SYSTEM ZONES DEFINITION *

ZDR (0,33675)? 800
'0BE9F....'0BB80

Ces tables sont constituées de pavés de 128, 10, 65, 36, 10 mots respectivement et de la ZDR.

ZUEP 128*(03, 32 MAX)? 4
'0BB7F....'0B980

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.

ZIOCB 10*(03, 32 MAX)? 4
'0B97F....'0B958

Ces questions n'apparaissent que si on est en fonctionnement sur 16-35 (macro %MODPRI absente)

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

ZWCB 65*(02,127 MAX)? 3
'0B957....'0B895

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

ZDF 36*(07,512 MAX)? 100
'0B894....'0AAB5

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

Sur Solar 16-70 cette zone est remplacée par la ZDM.

ZGIN 10*(03,384 MAX)? 50
'0AAB4....'0AB91

Zone utilisateur : Attention il faut configurer avec une taille N + 2 pour disposer de pavés de N mots.

ZWB (00,128 MAXI)? 5
B SIZE.....(0000,32767) 64
'A890....'0A751

7-2

```

*****
*           NUMBER OF TASKS           *
*****

```

```

ZBCT 38*(01, 120 MAX ) ? 26
'0A750....'0A375
*****

```

Pour chaque tâche, RTES16 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...

```

*****
*           PARTITIONS DEFINITION     *
*****

```

```

PARTITION START ADDRESS (MINIMUM '3B10)? ~4000

```

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).

```

MEMORY SIZE IN KWORDS      ? 256

```

Taille mémoire du calculateur sur lequel doit s'exécuter le système. Si le système, lorsqu'il est lancé, s'aperçoit que la mémoire est plus petite que celle indiquée, il émet un message sur DO ; il reste exploitable à condition de ne pas intégrer de tâche dans les partitions qui se trouvent au delà de la mémoire physique.

P00 ('10,'6370) ? '2000
'04000....'05FFF

P01 ('10,'4370) ? 512
'06000....'061FF

P02 ('10,'4170) ? '800
'06200....'069FF

P03 ('10,'3970) ? '800
'06A00....'071FF

P04 ('10,'3170) ? '3170
'07200....'0A36F

P05 ('10,'FFFF) ? '2000
'10000....'11FFF

P06 ('10,'E000) ? '4000
'12000....'15FFF

P07 ('10,'A000) ? '4000
'16000....'19FFF

P08 ('10,'6000) ? '1000
'1A000....'1AFFF

P09 ('10,'5000) ? '5000
'1B000....'1FFFF

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 1 est la première partition réservée aux programmes non résidents.

Lorsque cette configuration est conversationnelle (SI 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.

7-4

TASK NUMBER TO BE ACTIVATED ON RESTART OR 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 RTES16, RTES16 éditera le message : ERC 02 (tâche inexistante)

Si la tâche précisée est connue de RTES16 au moment du RESTART ou INIT elle est activée avec pour paramètre d'appel :

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

FILE NAME TO BE CALLED ON RESTART OR INIT..?INIT-ZZ

Le fichier indique doit être sur D2.

En l'absence de réponse DI sera affectée à TK.

Au moment du restart ou de l'INIT, RTES16 affecte DI au fichier indiqué qui doit se terminer par DI TK.

Si ce fichier n'existe pas DI est affecté à TK.

FIRST CDA PARTITION.....78
LAST CDA PARTITION.....78

7-5

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.

OK ? Y

Validation de la configuration.

La carte mémoire est alors réécrite dans le fichier image mémoire du système.

CONF16 rend le contrôle au moniteur.

*

- Système téléchargeable :

Il faut maintenant intégrer les tâches de l'application (TASK, EXEC) et frapper FCON pour valider la carte mémoire (voir la notice TELEC16)

- Autres types de système : on peut intégrer le système dans D1 par la commande GSYS.



Exemple de fichier CONF-SY correspondant à la carte mémoire des pages précédentes

```
10
2
800
4
4
3
100
50
5
64
26
4000
256
'2000
512
'800
'800
'3170
'2000
'4000
'4000
'1000
'5000
25
INIT-ZZ
8
8
Y
```



Utilisation des tables du système

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

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

ZUEP : 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.

ZIOCB : 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 N_1 est le nombre de pavés de la ZUEP et

N_2 le nombre de pavés de la ZIOCB, le degré de réentrance dans FMS est croissant avec : $\text{Inf}(N_1, N_2)$.

(ZWCB) : 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.

Sur un SOLAR 16-70, le nombre de ZUEP, ZIOCB, ZWCB n'est pas configurable par CONF16. En effet, ces pavés étant utilisés essentiellement par FMS, ils sont configurés avec la génération de FMS par les macro-instructions % DISK et % USER (voir la génération de FMS dans le manuel d'utilisation de FMS16).

ZDM : Zone de Manoeuvre

Cette zone de pavés remplace la ZDF sur un SOLAR 16-70. Elle contient :

- les ressources usager définies par RESDEF,
- des images d'IOCB émis par le mode privilégié.

Le nombre d'IOCB doit être au moins égal au niveau de réentrance de FMS (macro % USER).

Z D F : Zone des Descripteurs de Fichiers

Cette zone est remplacée par ZDM sur Solar 16-70 (voir ci-dessus)

Le système enregistre dans cette zone :

- les unités d'accès aux fichiers ouverts à un instant donné (16-35),
- les descripteurs de fichiers ouverts à un instant donné (16-35),
- les tables de ligature de granules des fichiers ouverts en accès direct rapide (16-35),
- 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" (16-35),
- 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) (16-35).

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 RTES16 (commande EXEC)
- un pavé lors de chaque exécution des requêtes de segmentation BCHLR 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.

ZWB : Zone des "Working Blocs"

Zone de travail pour l'utilisateur. Le nombre et la taille de ces pavés sont configurables. La requête GESPAV permet l'acquisition ou la libération d'un de ces pavés avec la possibilité de ne pas être bloqué lorsqu'il y a absence de pavés libres.



| | | |
|-----|---|------------------------------|
| - 5 | | |
| - 4 | RESERVE | |
| - 3 | DAP | |
| - 2 | | |
| - 1 | N° PRIORITE | |
| 0 | R A | |
| // | | |
| 11 | SLE | |
| 12 | NUMERO D'APPEL | NUMERO USAGER |
| 13 | NUMERO DE PARTITION OCCUPEE | TYPE ET PRIORITE |
| 14 | RESERVE TRAITEMENT ERREURS SUR REQUETES ET ALARMES | |
| 15 | ADRESSE PAVE "OVERLAY" ou '0000 | |
| 16 | NOM DU | |
| 17 | FICHIER | |
| 18 | | |
| 19 | "PUBLIC WORD" | |
| 20 | TAILLE IMAGE MEMOIRE | |
| 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 | EDON | 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 RTES16, 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

```
*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
```

8.2 - LANCEMENT DU MONITEUR "BACKGROUND"

```
*BACK, 9
  MAX JOB SIZE : 32 K WORDS
*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        P07             00548     TASK1  -AA   14
  017     S,017     N        P05             00063     TASK7  -AA   14
  018     S,018     N        P06 P05        00063     TASK8  -AA   14
  020     S,020     N        P09 P08 P07    00075     TASK10 -AA   14
  021     S,021     N        P07 P06        00067     TASK11 -AA   14
  022     S,022     N        P06             00075     TASK12 -AA   14
  025     S,025     N        P08             00182     TOVER  -AA   14
```

↓

Partitions accessibles

↓

Taille en nombre de mots (décimal)



*SYST, 1

| | @BEGIN | @END | MAX USED | NB USED | NB TOTAL |
|-------|--------|-------|-------------|-------------|--|
| ZUEP | 0B980 | 0BB7F | 002 | (000 / 004) | |
| ZIOCB | 0B958 | 0B97F | 001 | (000 / 004) | |
| ZWCB | 0B895 | 0B957 | 001 | (000 / 003) | |
| ZDF | 0AA85 | 0B894 | 010 | (006 / 100) | |
| ZGIN | 0A891 | 0AA84 | 003 | (002 / 050) | |
| ZBCT | 0A375 | 0A750 | 008 | (008 / 026) | |
| ZWB | 0A751 | 0A890 | 000 | (000 / 005) | |
| ZDR | 0BB80 | 0BE9F | | | |
| P00 | 04000 | 05FFF | S,126 | NT=122 | BACKM -:S |
| P01 | 06000 | 061FF | | | |
| P02 | 06200 | 069FF | | | Partition réservée au moniteur "background" |
| P03 | 06A00 | 071FF | | | |
| P04 | 07200 | 0A36F | | | |
| P05 | 10000 | 11FFF | | | |
| P06 | 12000 | 15FFF | | | |
| P07 | 16000 | 19FFF | | | |
| P08 | 1A000 | 1AFFF | | | |
| P09 | 1B000 | 1FFFF | | | |

*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

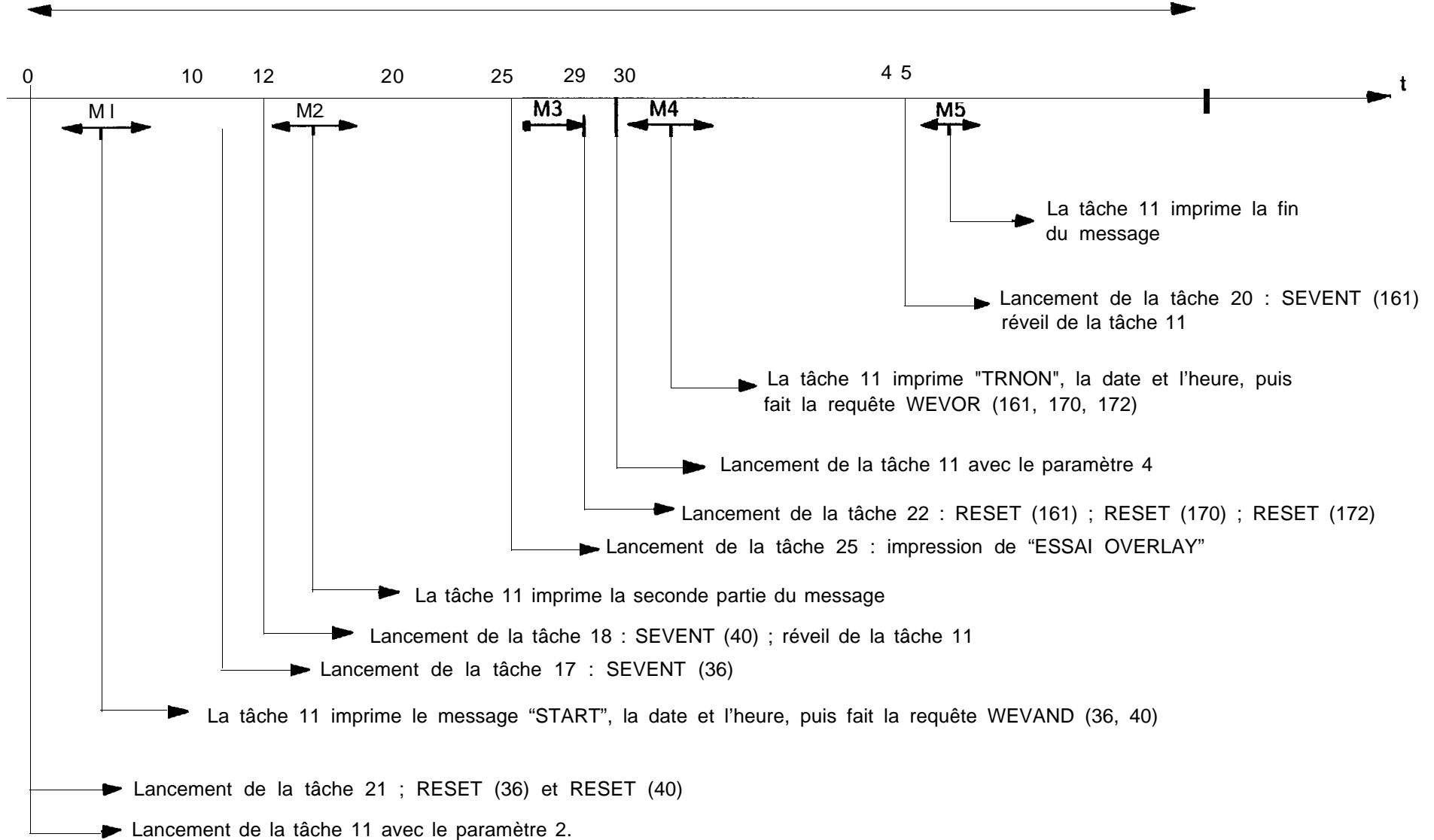
*TASK, R, TACRES-AA, 3

*SAVE, 3, 3

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

8.5 - Diagramme de temps

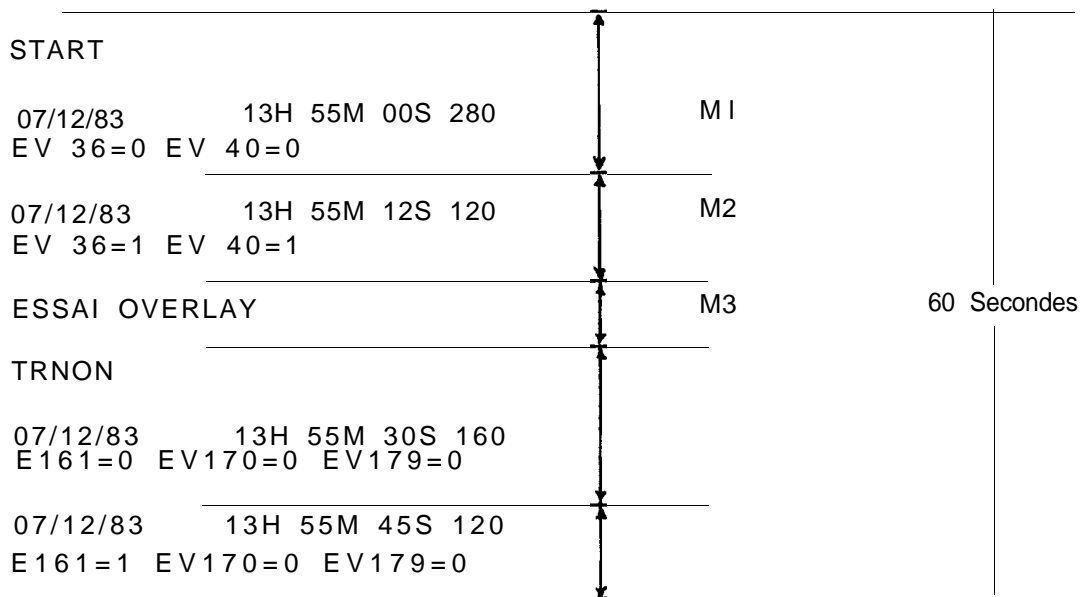
PERIODE 60 SECONDES





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/ ,13/54/45 | ----- | Initialisation de la date et de l'heure : le jeu sera lancé dans 15 secondes |
| *DOTS | | |





START

07/12/83 13H 56M 00S 200
EV 36=0 EV 40=0

07/12/83 13H 56M 12S 120
EV 36=1 EV 40=1

ESSAI OVERLAY

TRNON

07/12/83 13H 56M 30S 200
EV161=0 EV170=0 EV179=0

07/12/83 13H 56M 45S 120
EV161=1 EV170=0 EV179=0

START

07/12/83 13H 57M 00S 240
EV 36=0 EV 40=0

07/12/83 13H 57M 12S 100
EV 36=1 EV 40=1

ESSAI OVERLAY

TRNON

07/12/83 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.

RTES16 offre deux possibilités de RESTART :

- restart avec rechargement du système,
- restart sans rechargement du système.

9.1 - RESTART AVEC RECHARGEMENT

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 précise 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.



9.2 - RESTART SANS RECHARGEMENT

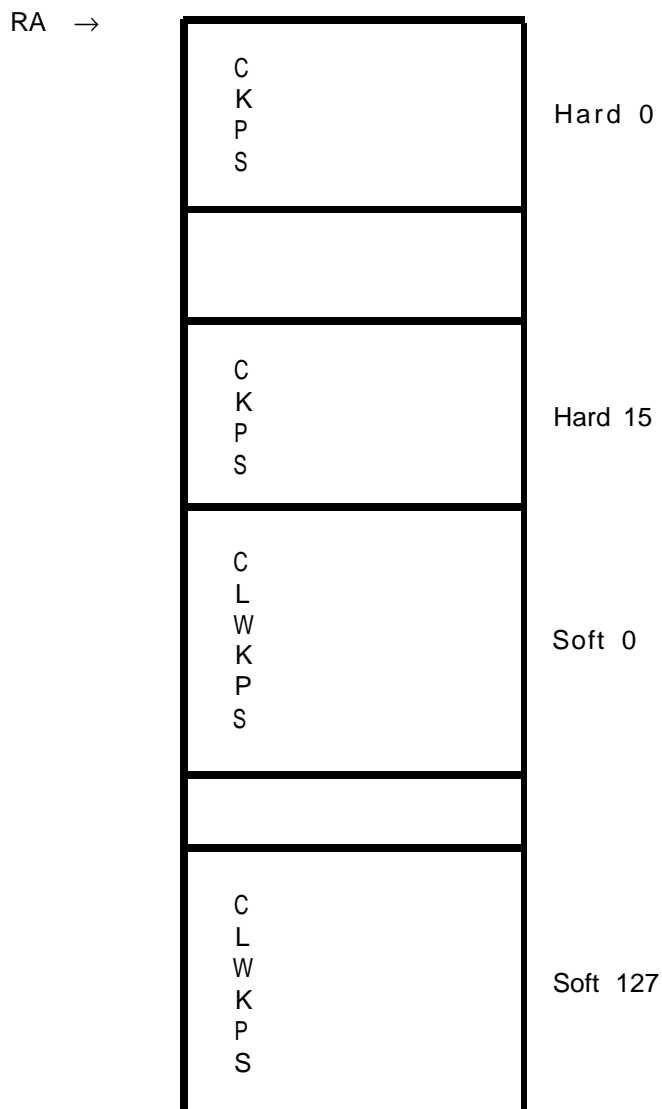
A la reprise secteur, le système est relancé sans rechargement du système ni de l'application. Le système effectue les opérations suivantes :

- Appel d'un sous-programme utilisateur
- Réinitialisation du système
- Réinitialisation de l'application
- Lancement d'une tâche utilisateur
- Lancement fichier de commande.

9.2.1 - Sous-programme utilisateur

Ce sous-programme est appelé dès la reprise secteur.

- Paramètres d'appel : RA = SLO zone de sauvegarde des contextes initiaux (elle se trouve en fond de mémoire) ; cette zone a la configuration suivante :



- Ce sous-programme (facultatif)
 - . doit être en mode maître
 - . ne doit pas faire de SVC ni démasquage
 - . doit se terminer par RSR
 - . pour être appelé, l'application doit mettre en RTCOM + '9C l'adresse du sous-programme
- Son rôle est de mémoriser les informations nécessaires au redémarrage de l'application.

9.2.2 - Réinitialisation du système

RTES16 réinitialise :

- les contextes des tâches système,
- les mémoires débanalisées et les files du scheduler,
- les pavés temporaires qui sont libérés (opérations sur horloges, pavés Run, événements, ...),
- les événements,
- les affectations SU-FU de la manière suivante :
 - . si affectation SU-FU : elle est conservée
 - . si affectation SU-fichier on remet l'affectation standard
- IOCS et FMS : tous les fichiers sont fermés.

9.2.3 - Réinitialisation de l'application

La réinitialisation consiste pour chaque tâche à :

- Mettre à jour le contexte de la tâche avec les valeurs de C, L, W, K, P, S initiales (sauvegardés au moment du TASK)
- Réinitialiser les files du scheduler en fonction de la nature de la tâche
- Remettre à 0 le nombre d'appels
- Réouvrir le fichier si nécessaire (tâche résidente avec overlay)
- Recalculer la valeur des registres SLO-SLE.

9.2.4 - Lancement de la tâche utilisateur

La tâche restart, si elle existe, est lancée avec un paramètre qui précise le type d'activation.

Paramètre = - 1 : Activation suite à une commande INIT ou à un INI pupitre

Paramètre = 0 : Restart avec rechargement

Paramètre = 1 : Restart sans rechargement



ANNEXE

Exemple de macro-instructions permettant de générer un RTES16 pour un Solar 16-70.

```
%FUGENE D2
%MODPRI
%BRANCHES RESIDENTES
%MACRO IOCS ON MACIOC-N3,D2 FMS ON MACFMS-N3,D2
%BUILD RTESN3 AVEC BACKGROUND FIN RACINE FF80
%BIBDRV ON BTRA16-:S,D2
%FIN
*END
```

Exemple de macro-instructions pour un 16-35

```
%FUGENE D2
%MACRO IOCS ON MACIOC-N3,D2 FMS ON MACFMS-N3,D2
%BUILD RTES35 AVEC BACKGROUND FIN RACINE FF80
%BIBDRV ON BTRA16-:S,D2
%FIN
*END
```



```
<=====
<          M A C R O S      I O C S      R T E S 1 6
<=====
ZUC 16-70
ZRTES16
<
<=====
<          N I V E A U  2
<=====
ZNIVEAU 2 KSTOR=
<
<
ZCPMUX4 MODE=LDC ADR='1900 IOP=  CONNEX=MUX4
ZPUMUX4 ASY STD SNIV1=08 SNIV2=12 ITN=28 VOIE=0
ZFU 27 CDE='0001 SENS=0 FINDIC=0
ZFU 28 CDE='0001 SENS=I FINDIC=1
ZPUMUX4 ASY STD SNIV1=09 SNIV2=13 ITN=29 VOIE=1
ZTUP-3=3
ZFU 22 CDE='0001 SENS=IO FINDIC=1
ZPUMUX4 ASY STD SNIV1=10 SNIV2=14 ITN=30 VOIE=2
ZTUP-3=3
ZFU 23 CDE='0001 SENS=IO FINDIC=1
ZPUMUX4 ASY STD SNIV1=11 SNIV2=15 ITN=31 VOIE=3
ZFU 24 CDE='0001 SENS=IO FINDIC=1
<
ZCPMUX4 MODE=LDC ADR='1908 IOP=  CONNEX=MUX4
ZPUMUX4 ASY STD SNIV1=00 SNIV2=04 ITN=20 VOIE=0
ZFU 25 CDE='0001 SENS=IO FINDIC=1
ZPUMUX4 ASY STD SNIV1=01 SNIV2=05 ITN=21 VOIE=1
ZFU 26 CDE='0001 SENS=IO FINDIC=1
<=====
<          N I V E A U  13
<=====
ZNIVEAU 13 KSTOR=
<
ZHTR MFI
<
<
ZCPMPX MT SNIV=3 MODE=HDC ADR='820 ITN=3 CONNEX= IOP= NBV=1
ZPUMPX MT VOIE=0
ZFU 9 CDE='0000 SENS=IO FINDIC=1
```



```
<=====
<
<                               NIVEAU 14
<=====
ZNIVEAU 14 KSTOR=
<
ZCR
ZLP
<-----
ZPUVM SNIV=1 MODE=HDC ADR='30 ITN=1 IOP=0 NBV=2
<
<----- UNITE 0 : MOBILE
ZFUIVM E1 VOIE=0 FIXE=N
ZFUESPVM D1
ZFUESPVM D2
<----- UNITE 0 : FIXE
ZFUIVM E2 VOIE=0 FIXE=Y
ZFUESPVM D3
ZFUESPVM D4
<----- UNITE 1 : MOBILE
ZFUIVM E3 VOIE=1 FIXE=N
ZFUESPVM D5
ZFUESPVM D6
ZFUESPVM D7
<----- UNITE 1 : FIXE
ZFUIVM E4 VOIE=1 FIXE=Y
ZFUESPVM D8
<----- DISQUE SMD (300 MO)
ZCPSMD SNIV=2 ADR='850 ITN=2 IOP=1
ZFUISMD E5 UNIT=0 FIXE=N
ZFUESPSMD E6
ZFUESPSMD E7
ZFUESPSMD E8
ZFUESPSMD E9
ZFUESPSMD EA
ZFUESPSMD EB
ZFUESPSMD ED
ZFUESPSMD EE
ZFUESPSMD EF
ZFUESPSMD D8
<
<=====
<
<                               NIVEAU 15
<=====
ZNIVEAU 15 KSTOR=
<
ZMFI ASY TTY
ZOPT ASY NBNULS=3 CRASY=N
ZFU 2 CDE='0000 SENS=0 FINDIC=0
ZFU 3 CDE='1000 SENS=I FINDIC=1
<
ZENDGEN
*EOT
```



```
%USER NB=4
%DISK NB=3
%ZDF NB=100
%PUFMS FUI=E1 OFU=0 NBFMSMAX=1 PUNBGRAN=2000
%PUFMS FUI=E2 OFU=0 NBFMSMAX=3 PUNBGRAN=3600
%PUFMS FUI=E3 OFU=0 NBFMSMAX=1 PUNBGRAN=1200
%PUFMS FUI=E4 OFU=0 NBFMSMAX=2 PUNBGRAN=2000
%PUFMS FUI=E5 OFU=1 NBFMSMAX=3 PUNBGRAN=12000
%EMA
%SEQ
%IND
%DIT
%SIX
%ALACARTE
*END
```


VOS REMARQUES SUR CE DOCUMENT

TITRE _____
RTES16 ADDENDUM A

N° DE REFERENCE _____
Bull-Sems : 1 164 324 01 030 04

DATE _____
JANVIER 1986

ERREURS DETECTEES _____

AMELIORATIONS SUGGEREES _____

- → Vos remarques et suggestions seront attentivement examinées.
si vous désirez une réponse écrite, veuillez indiquer ci-après
votre adresse postale complète.

NOM : DATE

SOCIETE :

ADRESSE :

- → Remettez cet imprimé à un responsable Bull-Sems ou envoyez le
directement à