division petits ordinateurs
et applications systèmes

COMPAGNIE INTERNATIONALE POUR L'INFORMATIQUE

mitra 15

manuel d'utilisation

Moniteur MultiTâches — MMT

Gamme	:	MITRA 15
Systèmes	:	MMT
Objet	:	Ce manuel décrit le moniteur multi tâches MMT disponible sur l'ordinateur MITRA 15. On y trouve les principes de base exposés par rapport aux principes de base du MTRD, la descriptio des commandes spécifiques et l'utilisation des modules moniteur.
Remarques	:	Version 5
Nombre de pages	:	44
Date d'édition	:	AVRIL 1975

Pour commander ce document, envoyez votre demande à l'adresse ci-contre en reproduisant intégralement la "référence document".

Compagnie Internationale pour l'Informatique
CIDOC 68, route de Versailles 78430. LOUVECIENNES

Référence document
4114 U1/FR

Moniteur MultiTâches — MMT

manuel d'utilisation

Sommaire

4114 U1/FR
44 pages

1. Introduction

1-1. Généralités

MMT (Moniteur Multi Tâches) se caractérise comme étant un sur-ensemble du moniteur MTRD (Moniteur Temps Réel Disque).

Le développement du MMT a été fait en assurant la compatibilité avec le MTRD et la tenue des performances du MTRD pour la prise en compte des tâches immédiates.

MTRD peut en fait être directement extrait du MMT par assemblage conditionnel.

Les extensions du MMT par rapport au MTRD peuvent se répartir en deux grandes catégories :

- fonctions qui étendent la notion de tâche et les moyens de coordination des tâches entre elles,
- fonctions qui permettent une meilleure utilisation de la mémoire centrale.

Le MMT est présenté dans ce manuel selon deux points de vue successifs :

- les principes de base, exposés par rapport aux principes de base du MTRD (chapitres 2 et 3),
- les spécifications détaillées des commandes opérateur et des appels superviseurs fournis à l'utilisateur (chapitres 4 et 5).

La lecture préalable du manuel d'utilisation du MTRD (4117 U/FR) est fortement conseillée.

1-2. Fonctions communes au MMT et au MTRD

MMT et MTRD présentent un certain nombre de fonctions communes. On peut citer :

1-2.1. Contrôle du dialogue opérateur

A tout instant l'opérateur, en provoquant une interruption, peut agir sur le moniteur par le biais de commandes paramétrées, frappées à la console de service.

- Commandes de lancement et de contrôle d'exécution des tâches.
- Commandes permettant de préciser les périphériques utilisés.
- Commandes d'aide à la mise au point des programmes au niveau 0.

1-2.2. Gestion centralisée des entrées/sorties

Une tâche accède à un périphérique à travers une étiquette opérationnelle ; la correspondance entre celle-ci et le périphérique est définie dans une table du moniteur qui peut être modifiée par une commande opérateur ou dynamiquement par l'appel à un module superviseur (M : ASGN).

Le traitement d'une entrée/sortie met en jeu les modules M : IO, M : IO2 et le handler du périphérique.

Le module M : IO réalise le contrôle logique de l'opération et, en fonction de l'assignation de l'étiquette opérationnelle utilisée, sélectionne le handler associé au périphérique désigné.

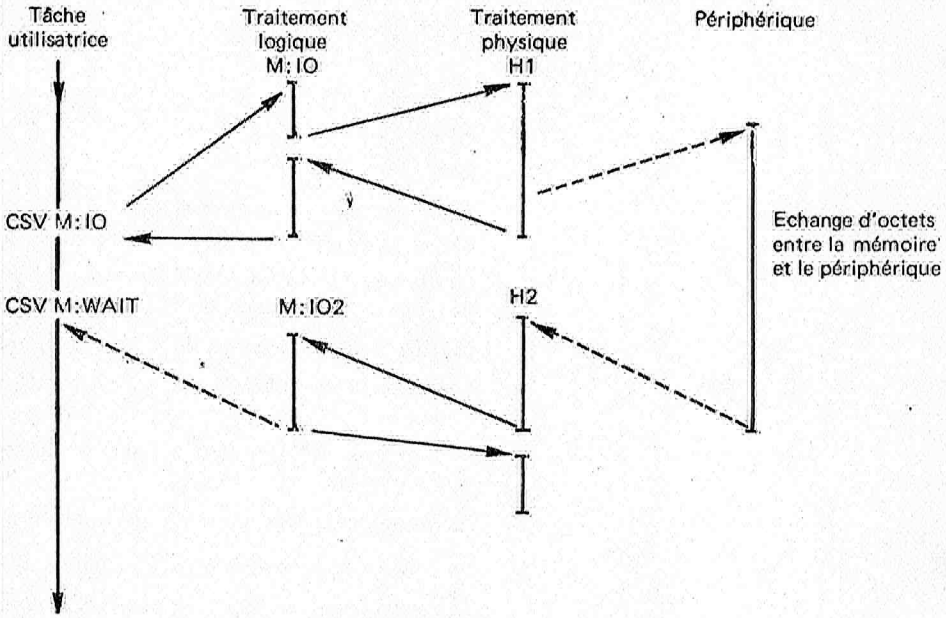
Le handler effectue le lancement et le contrôle physique du transfert.

Le module M : IO2 est appelé par la partie du handler qui traite l'interruption de fin de transfert pour réaliser la partie logique et générale du traitement.

Lors d'une demande d'entrée/sortie, alors que le périphérique correspondant est occupé à un premier échange, le module M : IO range la demande dans une file d'attente et rend le contrôle à l'utilisateur.

L'utilisateur peut mettre sa tâche en attente de la fin de l'échange par appel au module M : WAIT.

L'utilisateur peut demander le contrôle complet des erreurs de transfert.



1-2.3. Gestion des déroutements

- Sur des erreurs de fonctionnement du type adresse inexistante, code instruction invalide, violation de mode, ... le microprogramme réalise automatiquement l'appel du module M : TRAP qui édite sur la console de service les renseignements nécessaires à l'analyse de la cause de déroutement. La tâche ayant provoqué le déroutement est abandonnée.
- Lorsque le déroutement est provoqué par l'utilisation d'une instruction valide dont le module d'exécution (microprogramme ou logique câblée) n'existe pas dans la configuration, le module M : TRAP simule l'exécution de l'instruction ; le déroutement est alors transparent à l'utilisateur.

1-2.4. Contrôle du déroulement des programmes

En cas de déroulement défectueux d'une tâche (déroutement, erreur d'entrée/sortie) le moniteur, après édition d'information concernant cette anomalie, peut décider d'abandonner la tâche (ABORT).

Une tâche se termine normalement par un appel du module M : EXIT qui désactive la tâche.

1-2.5. Sous-programmes de conversion

Les sous-programmes de conversion utilisés par le moniteur sont réentrants et peuvent être appelés par l'utilisateur. Ils réalisent les conversions suivantes :

Binaire ⟷ chaîne hexadécimale
Binaire ⟷ chaîne décimale
ASCII ⟷ EBCDIC

1-2.6. Aide à la mise au point de programme

Des commandes opérateur permettent de contrôler l'exécution d'un programme ; elles réalisent les fonctions suivantes :

- dump d'une zone mémoire,
- arrêt sur une instruction avec édition des registres,
- trace d'une ou plusieurs instructions,
- dump d'une zone mémoire au passage sur une instruction,
- modification de données ou d'instructions dans un programme.

Ces commandes permettent une mise au point rapide des programmes, la mise au point au pupître restant toujours possible.

1-2.7. Gestion des programmes connectés à des interruptions

Le traitement des interruptions est en grande partie pris en compte par le microprogramme qui effectue la sauvegarde du contexte de la tâche en cours et le chargement du contexte de la tâche à lancer.

Le moniteur se charge des opérations suivantes :

- action sur les bascules d'interruption,
- connexion d'un programme à un niveau d'interruption prioritaire,
- abandon d'une tâche immédiate.

1-2.8. Gestion de la zone background

La zone background est une zone mémoire dans laquelle s'exécutent les tâches de plus basse priorité (priorité 0 matériel). Cette zone peut être utilisée de deux façons :

- Contrôle direct de l'enchaînement des programmes par commandes opérateur.
- Enchaînement séquentiel de travaux par le module BATCH. Les commandes sont alors contrôlées pour éviter toute action intempestive sur le reste du système et doivent être préparées sur un support externe (cartes, ruban, bande magnétique...). Le jeu d'étiquettes opérationnelles est spécifique.

1-2.9. Gestion du disque système

Une des unités de disque est gérée de manière spécifique par le moniteur.

Cette unité est divisée en plusieurs périphériques logiques correspondant à des zones de ce disque définies à la génération. A chaque zone est assignée une étiquette opérationnelle.

L'accès à ce disque se fait ainsi en adressage relatif au début de la zone choisie avec contrôle automatique de débordement.

1-2.10. Gestion de programmes en overlays

Le moniteur permet l'exécution de programmes possédant une structure de recouvrement.

Ce système permet de n'avoir en mémoire à un instant donné que les sections nécessaires, compte tenu de la structure des appels entre sections et de la fréquence d'utilisation de chacune d'elles.

L'appel d'une section provoque son chargement à partir du disque système si la section n'est pas en mémoire.

La structure arborescente du programme doit être définie par l'utilisateur à l'édition de liens.

1-2.11. Gestion de fichiers sur disque

La zone DA du disque système et les autres unités de disque servent à stocker les fichiers utilisateurs.

La gestion de ces fichiers peut être effectuée sous le contrôle du Système de Gestion de Fichiers disque SGF 15. Les accès sont faits par le module M : IO à travers les étiquettes opérationnelles.

Les modes d'accès autorisés sont :

- accès direct,
- accès séquentiel non bloqué,
- accès séquentiel bloqué.

1-2.12. Gestion de l'heure et des délais

La date et l'heure sont entretenues automatiquement dans le moniteur par une tâche immédiate activée par une horloge externe.

Ces éléments sont mis à la disposition de l'utilisateur par un appel au module M : TIME.

A la demande de l'utilisateur, le système peut mettre une tâche en attente de la fin d'écoulement d'un délai. La tâche sera automatiquement réactivée à l'échéance de ce délai.

Le MMT gère trois types de délais :

- délais simples,
- délais purs,
- time-out.

Les délais simples permettent de dissocier demande de début d'écoulement du délai de mise en attente. A chaque demande de délai simple est associé un bloc de contrôle (CB) contenant la valeur du délai demandé et un événement pour lequel la tâche peut demander sa mise en attente. L'événement sera activé à l'expiration du délai ou par une demande d'activation faite par une autre tâche.

Les délais purs n'ont pas d'événement associé : la tâche est automatiquement mise en attente au moment de la demande d'écoulement du délai.

Le "time-out" est un délai associé à une demande d'entrée/sortie. La valeur du délai est définie dans le control bloc d'entrée/sortie. La tâche sera réveillée par la fin d'entrée - sortie ou par la fin du délai, ce qui permet de contrôler la durée d'un échange.

1-2.13. Gestion de ressources

Pour faciliter le contrôle du partage des ressources telles que zone mémoire et périphériques, entre les tâches, le moniteur met à la disposition de l'utilisateur un ensemble de drapeaux qui permettent de matérialiser l'occupation d'une ressource par une tâche.

Une tâche qui veut partager une ressource avec d'autres tâches doit en demander l'affectation au moniteur, puis la libération de la ressource lorsqu'elle a terminé. Le moniteur se charge, lors de la demande d'une ressource, de mettre la tâche demandante en attente de libération de la ressource : une ressource ne peut être possédée que par une seule tâche à la fois.

Cette gestion de ressources peut être utilisée par exemple :

- pour la protection de tâches (mise à jour),
- pour le partage d'un tampon d'entrée/sortie.

1-3. Fonctions particulières au MMT

On distinguera :

- Les fonctions qui étendent la notion de tâche et les moyens de coordination des tâches entre elles.
- Les fonctions qui permettent une meilleure utilisation de la mémoire centrale (swapping et allocation dynamique en zone commune).

1-3.1. Extension de la notion de tâche

Sous MTRD une tâche est associée à la fois à un programme (logiciel) et à un niveau d'interruption (matériel). Le nombre maximum de tâches est ainsi limité par le nombre de niveaux d'interruptions disponibles dans le système utilisé, ce nombre étant lui-même limité à 32. Ce type de tâche est appelé "tâche immédiate" et existe évidemment sous MMT.

MMT introduit la notion de "tâche différée" : les tâches différées s'exécutent au niveau 1 qui est multiplexé par logiciel. Le nombre maximum de tâches différées existant à un instant donné est fixé à la génération du système et ne peut dépasser 94.

Les tâches différées peuvent être créées dynamiquement par appel au module M : TASK ; une priorité de prise en compte est fixée à la création.

Les tâches immédiates sont prises en compte suivant leur priorité matérielle, les tâches différées d'après leur priorité logicielle, les tâches immédiates sont toutes plus prioritaires que les tâches différées.

A la création la tâche est immédiatement prise en compte à moins qu'elle ne soit spécifiée en attente de délai ou d'événement.

A chaque modification de l'état des tâches (création d'une tâche, fin d'entrée/sortie, échéance d'un délai, libération d'une ressource,...) une ou plusieurs tâches qui étaient en attente deviennent prêtes à s'exécuter : on dit qu'elles sont éligibles. Le moniteur considère alors les priorités des tâches éligibles pour déterminer la tâche à laquelle il doit donner le contrôle.

La notion de tâche différée est indépendante de la notion de programme, plusieurs tâches différées pouvant s'exécuter sur le même programme.

1-3.2. Extension des moyens de coordination des tâches

Sous MMT la gestion des événements est centralisée. Les tâches en attente d'un même événement sont chaînées dans une file d'attente. Plusieurs tâches immédiates et/ou différées peuvent être mises en attente du même événement ; l'arrivée de l'événement rend toutes les tâches en attente de cet événement, éligibles.

En dehors de son application aux entrées/sorties et aux délais, la gestion des événements peut être utilisée pour synchroniser plusieurs tâches sur des événements purement logiciel : une tâche se met en attente d'événement par appel au module M : WAIT, une tâche active un événement par appel au module M : ACTV.

1-3.3. Swapping

MMT permet de créer des tâches différées à partir de programmes (IMT) stockés dans la zone EP du disque système. Les programmes sont chargés dynamiquement pour permettre de lancer la tâche. Ces programmes sont chargés dans une partition de la zone Middleground, nouvelle zone gérée par le moniteur. Les partitions sont fixées à la génération mais peuvent contenir une ou deux tâches.

Une tâche différée occupe la zone mémoire où elle a été chargée, soit jusqu'à sa fin normale, soit jusqu'à sa sauvegarde sur disque décidée par le moniteur après sa mise en attente : cette opération dite de "swapping-out" permet de libérer la place nécessaire à l'exécution d'une tâche plus prioritaire et éligible.

Cette fonction est normalement transparente pour l'utilisateur ; celui-ci peut toutefois demander le blocage d'une tâche différée en mémoire et la débloquer quand il le juge utile.

1-3.4. Allocation dynamique de mémoire en zone commune

Afin d'optimiser la gestion de la mémoire, une partie de la zone commune (zone commune dynamique) peut être allouée partiellement et momentanément aux tâches qui en font la demande. Les zones ainsi allouées ne sont pas "swappées" sur disque avec la tâche qui les a réservées.

Cette zone est essentiellement destinée à contenir des blocs de commande, des tampons d'entrée/sortie, des zones de travail et les paramètres à transmettre lors de la création de tâches différées.

2. Rappel des principes de base du MTRD

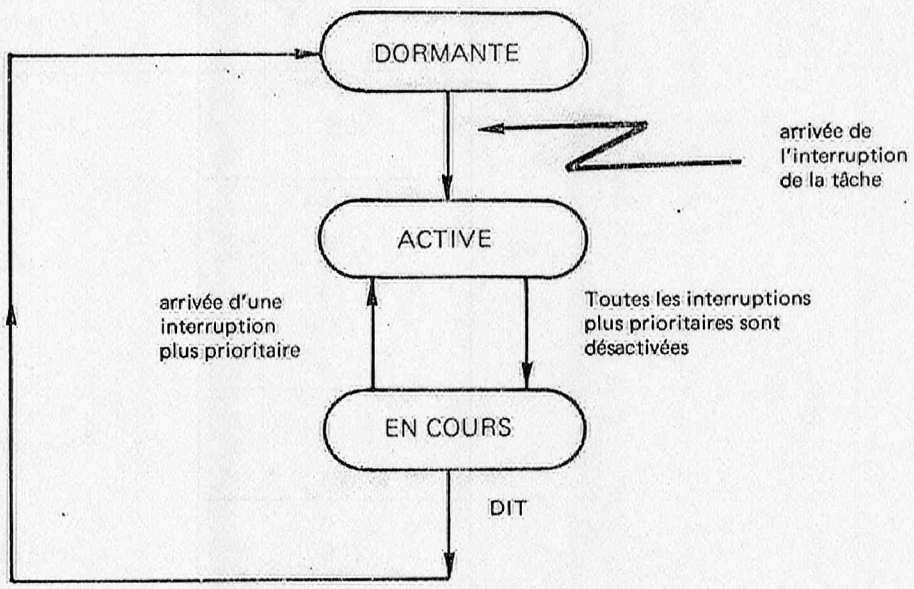
2-1. Principales caractéristiques du MTRD

On citera les principales caractéristiques suivantes :

- La multiprogrammation n'est assurée que par le dispositif matériel de priorité des interruptions.
- Chaque tâche est associée à un niveau d'interruption, c'est-à-dire qu'elle est associée à une priorité et à un programme. Le nombre total de tâches est ainsi limité au nombre d'interruptions du système, ce nombre étant lui-même limité à 32.
- Les tâches sont résidentes en mémoire avec toutefois la possibilité d'overlays.
- Les tâches ne sont coordonnées entre elles que par le jeu des interruptions.
- Les partitions mémoire sont fixées et gérées par l'opérateur.
- Les tâches peuvent communiquer entre elles par la zone commune dont la structure est figée.
- Les tâches sont identifiées par un numéro figé et dépendant du système (numéro d'interruption).

2-2. Etats d'une tâche vus par la micromachine

- L'état d'une tâche est gérée par le biais du contexte de la tâche ; ce contexte de 7 mots est pointé par la table CPT.
- Une tâche peut être "dormante", "active", ou "en cours" ; les transitions entre ces états sont automatiquement réalisées par la micromachine (scheduler matériel).



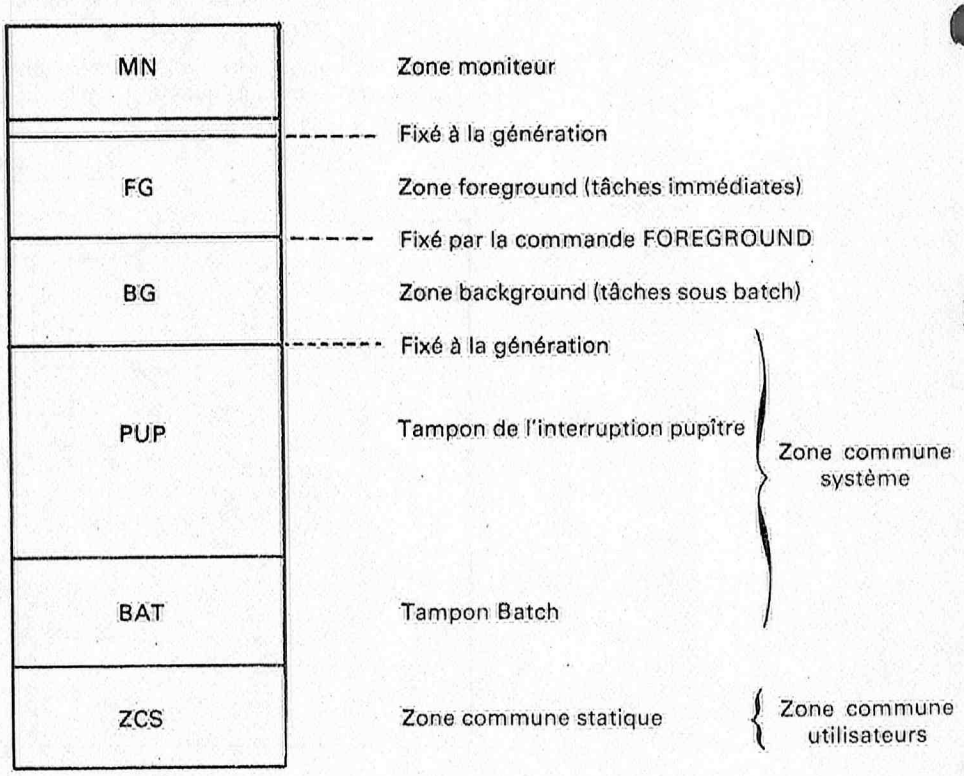
2-3. Principales fonctions sous MTRD

On retiendra notamment :

M : IT	Excitation d'un niveau : action sur armement et validation.
M : EXIT	Fin de tâche : désactivation du niveau.
M : KILL	Abort d'une tâche : déconnexion du niveau.
M : ABRT	Autodestruction d'une tâche : déconnexion du niveau.
M : WAIT M : ZWAT	Attente d'un événement.
M : IO M : ZIO	Demande de transfert.
M : DLAY M : ZDLY	Demande de délai.
M : SDLY M : ZSDL	Suppression de délai.
M : LOAD	Chargement d'un programme en mémoire.
M : BIB	Localisation d'un programme sur disque.
M : CNEC	Connexion d'un programme à un niveau.
M : RQST	Réservation d'une ressource.
M : RLSE	Libération d'une ressource.
M : TAR	Test d'une ressource.

2-4. Découpage de la mémoire pour MTRD

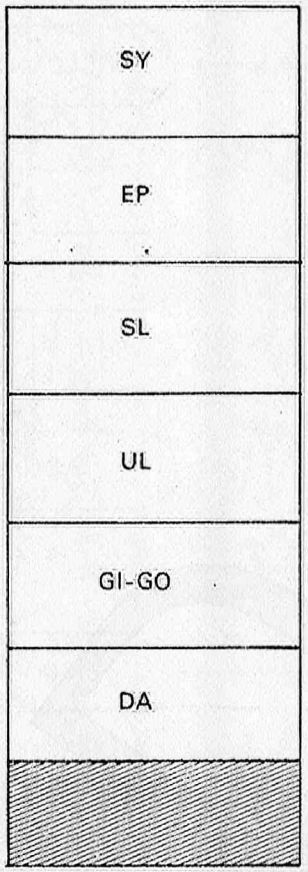
On distingue 6 zones :



2-5. Découpage du disque système pour MTRD

Une des unités de disque est gérée d'une façon spéciale par le moniteur : cette unité est divisée en plusieurs zones à la génération ; à chaque zone est associée de façon permanente une étiquette opérationnelle.

Des extensions de la zone DA peuvent exister sur d'autres unités de disque.



Zone moniteur (IMA)

Bibliothèque des programmes destinés à constituer des tâches (IMT)

Bibliothèque des modules système (BT)

Bibliothèque des modules utilisateur (BT)

Zone de travail des processeurs

Zone des données, éventuellement gérée par SGF 15

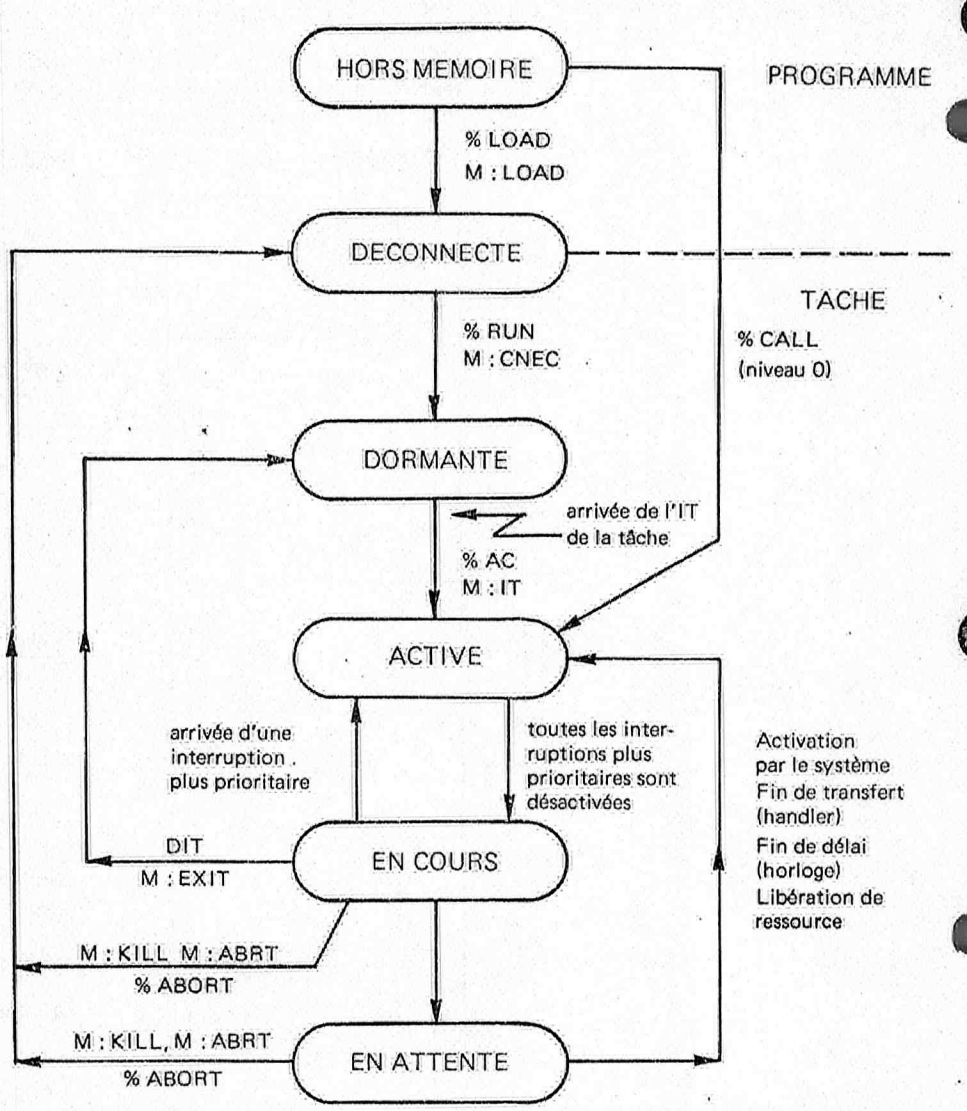
Fin physique du disque système

2-6. Etats d'une tâche vus par MTRD

La notion de tâche s'enrichit des notions de tâche "en attente", tâche "déconnectée" et il apparaît une différenciation entre tâche et programme.

La transition entre les différents états d'une tâche est contrôlée par le moniteur à l'exception de la transition entre état "actif" et état "en cours" qui est contrôlé par la micromachine. Une tâche peut être en overlays.

Le contexte est étendu à 8 mots par logiciel, le 8ème mot servant à gérer les chargements, abort et overlays.



2-7. Terminologie appliquée au MTRD

- **Tâche immédiate :** tâche associée à un niveau d'interruption. La tâche est identifiée par un numéro fixe dépendant du matériel : c'est le niveau de l'interruption.
- **Tâche foreground :** tâche immédiate.
- **Tâche background :** tâche associée au niveau zéro.
- **Chargement d'un programme :** introduction d'un programme en mémoire pour exécution éventuelle.
- **Création d'une tâche :** connexion à un niveau d'interruption d'un programme préalablement chargé.
- **Activation ou réveil d'une tâche :** excitation de son niveau d'interruption.
- **Suspension d'une tâche :** désactivation de son niveau d'interruption (retour à l'état dormant).
- **Destruction d'une tâche :** déconnexion du niveau d'interruption et connexion du niveau à une séquence DIT, BRU $-1.
- **Mise en attente d'une tâche :** désactivation du niveau en attente d'un événement ou de ressource.
- **Réactivation d'une tâche :** excitation du niveau après une mise en attente.
- **Priorité d'une tâche :** niveau d'interruption associé.

2-8. Dispositifs logiciel de changement d'état sous MTRD

— **Création**	% RUN M : CNEC
— **Activation**	% ACTIV M : IT
— **Suspension**	M : EXIT, DIT
— **Destruction**	% ABORT M : KILL, M : ABRT
— **Mise en attente**	M : IO + M : WAIT M : ZIO + M : ZWAT M : DLAY + M : WAIT M : ZDLY + M : ZWAT M : RQST (si la ressource est occupée par une autre tâche).
— **Réactivation**	. Arrivée de l'événement fin de transfert (handler). . Arrivée de l'événement fin de délai (interruption horloge). . Libération de la ressource par une autre tâche (M : RLSE).
— **Fin**	M : EXIT suivi d'un branchement pour réinitialisation.

2-9. Principales fonctions opérateur du MTRD

% LOAD	Chargement de programme.
% RUN	Connexion d'un programme à un contexte.
% CALL	Exécution en background.
% ACTIV	Excitation d'un niveau d'interruption.
% IT	Armement, validation d'un niveau.
% ABORT	Déconnexion d'un niveau.
% FOREGROUND	Déplacement de la frontière foreground/background.
% ASSIGN	Assignation d'un périphérique à une étiquette opérationnelle.
% TIME	Réinitialisation de l'horloge.

2-10. Traitement des événements par MTRD

Une seule tâche peut être en attente d'un même événement : à chaque événement est associé un mot (le mot événement d'un CB) ; pendant l'attente de l'arrivée de l'événement, le mot événement contient le numéro de la tâche en attente.

3. Principes de base du MMT

3-1. Généralités

Neuf objectifs principaux ont été choisis pour le développement du MMT :

- extension de la notion de tâche,
- possibilité de définir des tâches résidentes sur disque,
- coordination possible entre tâches,
- compatibilité avec MTRD,
- indépendance de chaînes de traitement entre elles,
- indépendance possible des tâches vis-à-vis du système sous lequel elles s'exécutent,
- préservation des performances temps réel du MTRD,
- mise en overlay de certaines fonctions moniteur,
- minimiser l'obligation pour une tâche de résider en mémoire.

3-2. Extension de la notion de tâche

3-2.1. Utilisation du niveau 1

Le niveau 1 est multiplexé par logiciel. Les tâches qui partagent le niveau 1 sont appelées "tâches différées". La table CPT des pointeurs de contexte est étendue par une gestion logiciel au-delà des 32 éléments correspondant aux interruptions, jusqu'à un maximum de 128. A un instant donné il peut donc exister un maximum ($\leqslant 94$) de tâches différées, mais l'allocation dynamique des éléments de CPT autorise les créations successives d'un nombre pratiquement illimité de tâches.

Le niveau 1 est hiérarchisé en priorités logiciel, plusieurs tâches différées pouvant avoir la même priorité.

Le niveau 1 est géré par un programme moniteur appelé SCHEDULER qui s'exécute lui-même au niveau 1.

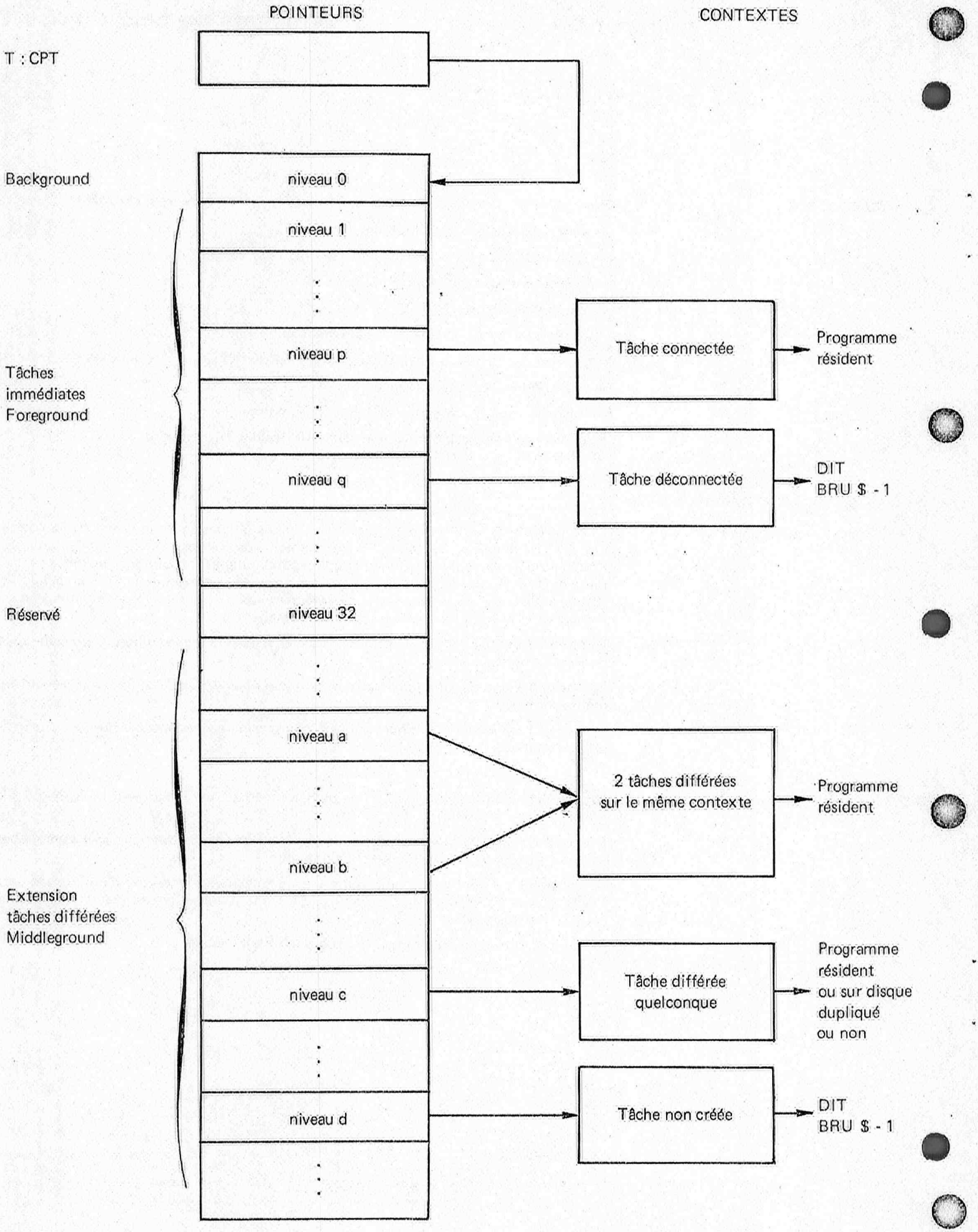
Le niveau 1 est évidemment géré par la micromachine par rapport aux autres niveaux de priorité matériel.

3-2.2. Indépendance tâche-programme

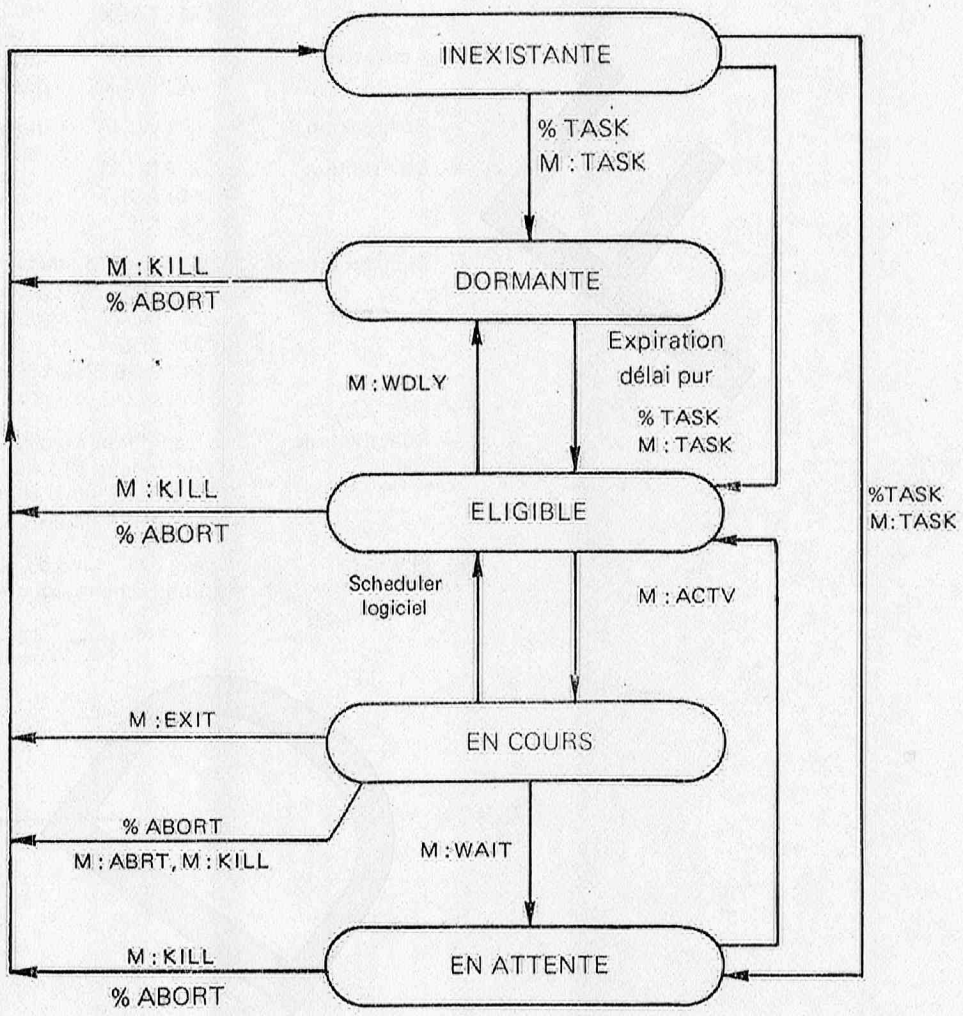
Un contexte unique est associé à une tâche immédiate par l'opérateur (% RUN) ou à la génération du système.

Un contexte peut être associé dynamiquement à une tâche différée ; plusieurs tâches peuvent s'exécuter sur le même programme :

- sur le même contexte : comme il n'y a aucune réentrance, une file d'attente sera créée pour l'utilisation du programme. Ce mode d'utilisation est réservé aux tâches différées résidentes,
- sur des contextes différents : le programme est recopié.


POINTEURS
CONTEXTES
T : CPT
Background
niveau 0
niveau 1
niveau p
Tâches
immédiates
Foreground
Tâche connectée
Programme
résident
niveau q
Tâche déconnectée
DIT
BRU $ - 1
Réservé
niveau 32
niveau a
2 tâches différées
sur le même contexte
Programme
résident
niveau b
Extension
tâches différées
Middleground
niveau c
Tâche différée
quelconque
Programme
résident
ou sur disque
dupliqué
ou non
niveau d
Tâche non créée
DIT
BRU $ - 1

3-2.3. Etats d'une tâche différée vus par le MMT



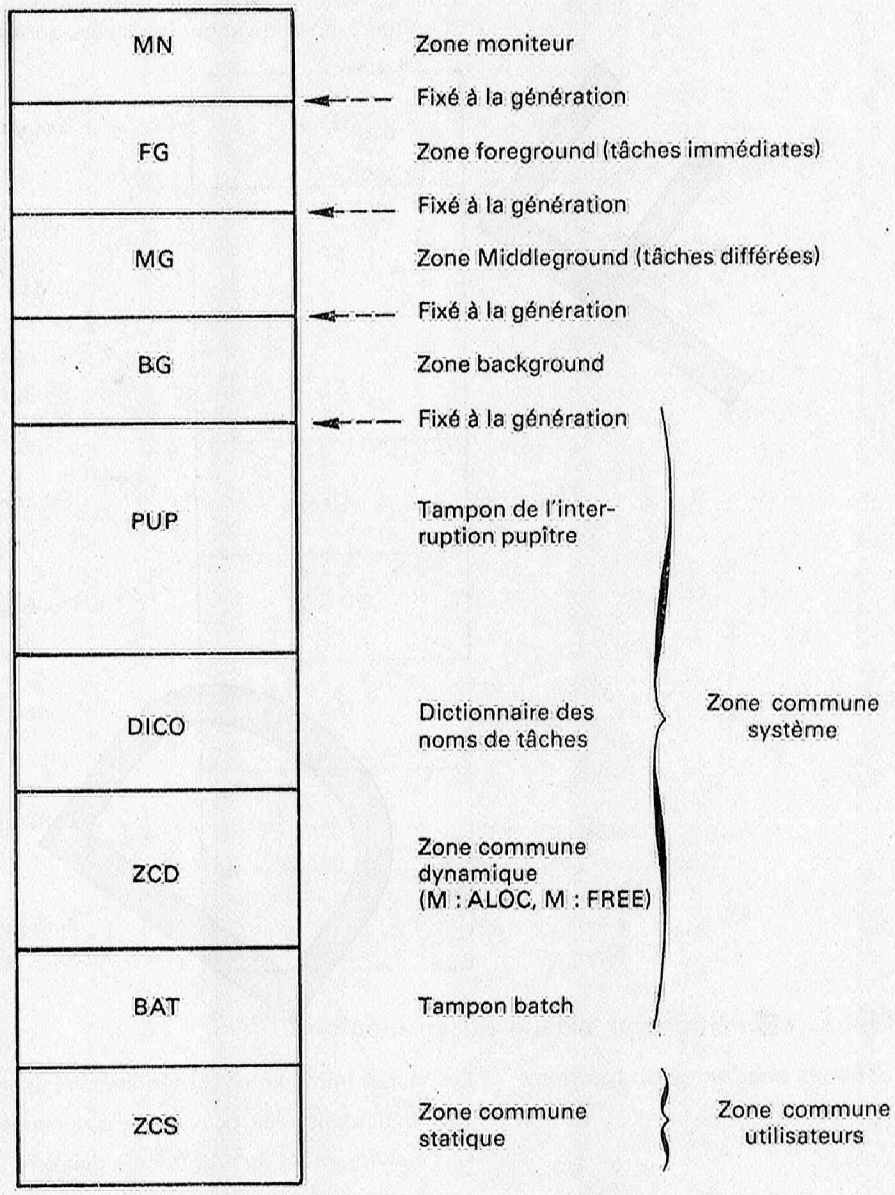
3-2.4. Terminologie appliquée aux tâches différées

- **Création d'une tâche :** allocation par le système d'un numéro de tâche différée et connexion du programme (en mémoire ou non) désigné par l'utilisateur.
- **Activation ou réveil d'une tâche :** rendre la tâche connue du Scheduler logiciel (la tâche est "Éligible").
- **Suspension d'une tâche :** rendre non éligible c'est-à-dire ignorée du Scheduler logiciel (la tâche est "Dormante").
- **Destruction d'une tâche :** déconnexion du programme et libération du numéro de tâche différée (la tâche est "Inexistante").
- **Mise en attente d'une tâche :** rendre non éligible et mettre dans une file d'attente dépendant de la nature de l'attente (la tâche est en "Attente").
- **Réactivation d'une tâche :** sortir une tâche de sa file d'attente et la rendre éligible.
- **Priorité d'une tâche :** elle désigne une des files d'attente de tâches éligibles connues du Scheduler logiciel.
- **Fin d'une tâche :** auto-destruction de la tâche. Cependant dans le cas d'une tâche résidente, le contexte est conservé de façon qu'on puisse de nouveau créer une tâche sur le programme correspondant ; il est entendu que l'utilisateur a prévu un branchement sur une séquence de réinitialisation dynamique.

3-2.5. Dispositifs logiciel de changement d'état d'une tâche différée

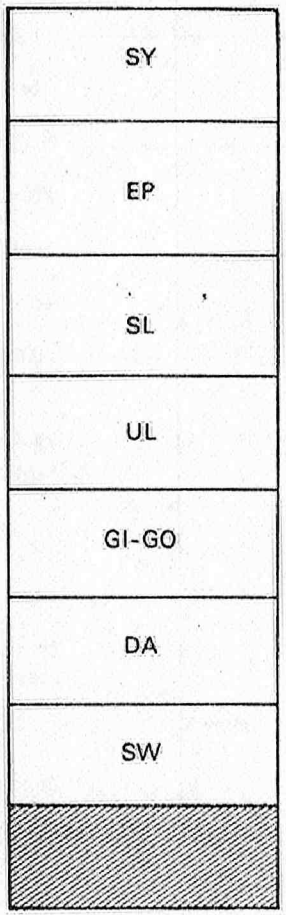
— **Création**	% TASK M : TASK
— **Activation**	% TASK (création et M : TASK réveil)
— **Suspension**	M : WDLY (attente de délai pur)
— **Destruction**	% ABORT M : ABRT M : KILL
— **Mise en attente**	M : IO + M : WAIT (attente fin de transfert) M : ZIO + M : ZWAT M : DLAY + M : WAIT (attente délai ou événement) M : ZDLY + M : ZWAT (attente délai ou événement) M : RQST (si la ressource est occupée par une autre tâche) M : ALOC (s'il n'y a plus de bloc disponible)
— **Réactivation**	Handler et M : ACTV Horloge et M : ACTV M : RLSE (libération de la ressource) M : FREE (libération d'un bloc)
— **FIN**	M : EXIT suivi d'un branchement pour réinitialisation si c'est une tâche résidente.

3-2.6. Découpage mémoire pour MMT On distingue 9 zones :



3-2.7. Découpage du disque système pour MMT

Comme pour MTRD une unité de disques est gérée d'une façon spéciale par le moniteur : on y distingue plusieurs zones :



Zone moniteur (IMA)

Bibliothèque des programmes destinés à constituer des tâches (IMT)

Bibliothèque des modules système (BT)

Bibliothèque des modules utilisateur (BT)

Zone de travail des processeurs

Zone des données, éventuellement gérée par SGF 15

Zone de swapping

Fin physique du disque système.

3-3. Tâches résidentes sur disque ou en mémoire

3-3.1. Tâches résidentes en mémoire

Les tâches immédiates sont toujours résidentes en mémoire.

Les tâches différées peuvent être déclarées résidentes en mémoire :

- Le chargement du programme doit être effectué par l'opérateur en zone Foreground.
- Le nom du programme est alors enregistré dans un dictionnaire résident en mémoire.
- Une file d'attente est associée au programme.

3-3.2. Tâches résidentes sur disque

Les programmes sur les quels on peut créer des tâches différées peuvent être déclarés résidents sur disque par catalogage en bibliothèque EP.

Le chargement de ces programmes est automatiquement effectué en zone Middleground au moment de l'activation des tâches correspondantes.

Ces tâches sont a priori swappables et elles peuvent avoir une structure en overlays.

3-3.3. Gestion de la zone Middleground

La zone Middleground est découpée en partitions fixes au moment de la génération. Chaque zone correspond à une priorité maximum (indice de la partition).

Les partitions sont allouées dynamiquement en fonction :

- de la disponibilité de la partition,
- de la taille du programme,
- de la priorité de la tâche.

Une tâche de priorité n peut être affectée à la partition d'indice n ou à la partition d'indice n-1 si on ne peut affecter la tâche à la partition précédente,...

Une tâche ne peut être affectée à une partition dont l'indice est supérieur à sa priorité :

- si la partition n'est pas disponible : une partition peut contenir au maximum 2 programmes (chargement par le bas et chargement par le haut).
- si la place mémoire qui reste disponible dans la partition est insuffisante.

Les partitions peuvent être rendues disponibles par swapping-out en fonction :

- de l'état de la tâche (en attente ou éligible),
- de la priorité de la tâche,
- de la taille mémoire libérée,
- du verrouillage éventuel de la tâche en mémoire.

Cette solution présente l'avantage de libérer la mémoire pour les tâches dont le temps de réponse est de l'ordre d'un transfert disque.

L'utilisateur peut demander le verrouillage d'une tâche en mémoire (et évidemment le déverrouillage) : cette technique doit en particulier être utilisée dans le cas de transferts en zone programme (intérêt de l'utilisation de la zone commune).

Tâche de priorité 0 :

Une telle tâche ne peut être chargée que seule dans son (unique) partition ; l'espace restant derrière elle lui est automatiquement attribué.

On peut ainsi, en l'absence de zone background, faire tourner les processeurs, implantés à adresse fixe, et réservant leurs tables derrière eux jusqu'à la zone commune. Ces tables sont swappées-out avec le reste de la tâche.

Scheduling réduit :

Le Scheduler met à profit les temps d'attente du swapping (exécuté au niveau 2) en lançant la tâche éligible en mémoire la plus prioritaire.

3-3.4. Cas où une tâche peut être swappée-out

Une tâche immédiate, une tâche différée résidente en mémoire et une tâche différée résidente sur disque mais verrouillée en mémoire ne peuvent pas être swappées-out.

Une tâche différée résidente sur disque et non verrouillée en mémoire **peut à tout moment** être swappée-out.

Le swapping sera réellement effectué dans le cas où les 2 conditions suivantes sont remplies :

- Le scheduler veut mettre en cours une tâche éligible de priorité supérieure.
- Cette tâche n'est pas en mémoire et le swapper ne trouve pas d'autre place en mémoire que la place occupée par la tâche qui sera swappée out.

Ces conditions peuvent évidemment être réalisées lorsque la tâche considérée se met en attente d'événement (entrée/sortie ou délai), de ressource ou de bloc dynamique de mémoire.

Ces conditions peuvent aussi se réaliser pour une tâche en cours d'exécution (et qui ne fait pas d'appel moniteur) si une interruption externe se produit ou si un événement arrive (fin de délai ou d'entrée/sortie), cet événement libérant notamment une tâche plus prioritaire.

3-4. Coordination entre tâches

3-4.1. Généralités

La coordination entre tâches peut se faire par le jeu des notions de ressource, d'événement et d'allocation dynamique de blocs mémoire.

La solution adoptée est celle d'un système de files d'attente qui prend en compte les tâches immédiates et les tâches différées. Cette solution entraîne un overhead système un peu plus important pour les tâches immédiates. Une tâche ne peut être que dans une seule file d'attente ; si elle n'est dans aucune file, elle est dans l'état dormant.

3-4.2. Événements

Un événement quelconque est matérialisé par un mot en zone programme ou en zone commune. La mise en attente d'une tâche sur événement se fait par appel au module M : WAIT ou M : ZWAT (zone programme ou zone commune). Le numéro de la tâche (ou niveau) est chaîné dans une file d'attente liée à l'événement.

L'activation d'un événement se fait par appel au module M : ACTV ou M : ZACT (zone programme ou zone commune). Cette activation débloque **simultanément** toutes les tâches en attente de cet événement (les tâches deviennent éligibles).

Pour coordonner un ensemble de tâches sur événement, la méthode normale est d'utiliser un événement en zone commune.

A tout événement peut être associé un événement en zone commune : l'activation du premier événement entraîne l'activation du second.

Toutes ces possibilités sont étendues aux tâches immédiates.

L'événement fin de transfert est automatiquement activé par le système de gestion des entrées/sorties, l'événement fin de délai est automatiquement activé par l'horloge.

3-4.3. Ressources

Une ressource est matérialisée par un numéro défini à la génération du système ; certains numéros sont réservés par le système (standard : 0-9).

La prise de possession d'une ressource se fait par appel au module M : RQST : cet appel est transparent si la ressource est déjà possédée par la tâche appelante ; dans l'autre cas, la tâche appelante est placée en file d'attente de ressource si la ressource n'est pas libre.

La libération de la ressource se fait par appel au module M : RLSE. Cet appel rend éligible une seule tâche : la tâche immédiate la plus prioritaire ou la première tâche différée en attente dans la pile.

Le test de l'état de la ressource se fait par appel au module M : TAR.

En cas d'abort d'une tâche, ses ressources sont libérées.

3-4.4. Allocation dynamique de mémoire

Un appel au module M : ALOC permet d'obtenir un bloc de mémoire de taille donnée dans la zone commune dynamique. Cette zone est en fait allouée par incréments de différents types.

Le nombre de types est déterminé à la génération. Le nombre de blocs d'un type peut varier dynamiquement par subdivision de blocs plus grands ou par concaténation de blocs plus petits.

Si un bloc du type demandé ne peut pas être obtenu la demande est mise en file d'attente.

Un bloc est libéré par un appel au module M : FREE. Le système effectue alors les subdivisions ou reconcaténations de blocs, et relance la tâche en tête de file d'attente du ou des blocs reconstitués.

Le test de disponibilité d'un bloc de type donné se fait par un appel au module M : TAL.

3-4.5. Files d'attente

Une tâche ne peut se trouver que dans une seule file d'attente à la fois.

Il existe plusieurs types de files d'attente :

- une file d'attente par priorité des tâches éligibles sur disque.
- une file d'attente par priorité des tâches éligibles en mémoire.
- une file d'attente par événement.
- une file d'attente par ressource.
- une file d'attente par type de bloc mémoire en zone commune dynamique.

Ces files d'attente sont du type FIFO à l'exception de la file d'attente d'événement qui est du type liste puisque toutes les tâches sont réactivées à l'arrivée de l'événement.

L'attente du délai pur est l'état dormant (hors de toute file).
Les tâches immédiates sont traitées d'abord et ceci par ordre de priorité.

3-5. Compatibilité avec MTRD

- MTRD sera désormais généré comme un sous-ensemble du MMT.
- Le traitement des tâches immédiates est totalement préservé ; seule la synchronisation de plusieurs tâches sur un même événement alourdit légèrement les traitements du moniteur.

- L'interface moniteur-utilisateur du MMT est un sur-ensemble de celui du MTRD.
- Tout programme s'exécutant sous MTRD peut s'exécuter sous un MMT qui respecte la structure de la zone commune statique.

3-6. Indépendance des chaînes de traitement

- Une chaîne de traitement est formée par une tâche mère, ses filles, ses petites filles, ...
- Une tâche connaît son propre numéro et les numéros de ses filles. On peut agir sur une tâche par divers appels utilisant le numéro de la tâche.
- On peut détruire toute une chaîne (ou sous chaîne) en détruisant la tâche mère. La destruction (abort) de la filiation est automatique.
- La gestion dynamique de la zone commune évite de partager de façon explicite la zone commune entre chaînes de traitement. Les adresses des blocs dynamiques peuvent être passés comme paramètres aux tâches d'une même filiation : on peut ainsi définir des éléments communs à la filiation et inconnus des autres chaînes.
- On notera toutefois que la chaîne de traitement doit prendre en charge le contrôle des saturations, c'est-à-dire la non disponibilité d'un numéro de tâche à la création.

3-7. Indépendance des tâches vis-à-vis du système

- Les tâches sont créées en spécifiant des noms de programme.
- On peut écrire un programme de façon à ce qu'il puisse s'exécuter à n'importe quel statut :
  - tâche immédiate,
  - tâche différée résidente en mémoire,
  - tâche différée résidente sur disque,
  - tâche background,
  - job sous batch.

Il suffit pour cela de la faire commencer par une séquence de réinitialisation des données locales.

- Si on n'exploite pas les numéros de tâche on peut laisser au système le traitement automatique des saturations.
- Il est possible d'intégrer dynamiquement des traitements dans le système en chargeant des programmes dans la zone EP du disque.
- L'allocation dynamique de blocs mémoire permet d'éviter une structuration statique de la zone commune.

3-8. Préservation des performances temps réel du MTRD

L'objectif recherché est de préserver les temps d'acquisition des interruptions (tâches immédiates) en n'imposant pas un temps de masquage des interruptions par le système supérieur à celui du MTRD.

Ce temps de masquage étant lié aux traitements des tables système qui sont plus importants que sous MTRD, la solution adoptée a été l'introduction de la notion de ressource système dont les séquences de prise de possession et de libération qui imposent le masquage, sont spécifiques et par suite très courtes.

Dans ces conditions une interruption peut être prise en compte avec un retard inférieur à 100 µs.

3-9. Mise en overlay de certaines fonctions moniteur

Pour réduire l'encombrement du moniteur, certaines fonctions sont mises en overlay : une demande de traitement est placée dans une pile ; un appel au niveau 2 lance la recherche et l'exécution des traitements par ordre de priorité.

Les fonctions en overlays sont :

- initialisation,
- réinitialisation,
- abort,
- gestion du swapping.

3-10. Minimisation de l'obligation de résidence des tâches en mémoire

2 types de cas imposent le verrouillage d'une tâche différée non résidente en mémoire :

- Demande d'entrée/sortie ou de délai simple en zone programme.
- Calcul d'adresses par rapport à G dans la zone commune.

Le verrouillage de la tâche en mémoire est automatiquement fait par le moniteur pour les cas du premier type. Ce verrouillage est toutefois évité si les demandes sont faites dans la zone commune.

Le verrouillage de la tâche en mémoire doit être fait par l'utilisateur pour les cas du second type. **Ce mode de programmation est à proscrire** et doit être remplacé par la méthode suivante :

- Calcul d'un pointeur relatif au début de la zone commune dans le registre X.
- LDA (ou STA) @ ≠ ZC, X avec ZC EQU 6

Le mot 6 de la CDS est en effet toujours maintenu par le moniteur et contient l'adresse par rapport à G du premier mot de la zone commune. La tâche peut ainsi être interrompue et swappée-out après chaque instruction.

4. Commandes opérateur

4-1. Commandes opérateur identiques à celles de MTRD

Ces commandes sont complètement décrites dans le manuel d'utilisation du MTRD.

Nous nous contenterons de rappeler leurs principales caractéristiques :

% RUN	Lancement du dernier programme chargé
% CALL	Chargement d'un IMT en background et lancement
% ABORT	Abandon d'un programme
% DISPLAY	Edition des éléments système
% DUMP	Edition mémoire
% ASSIGN	Assignation des étiquettes opérationnelles
% DEVICE	Action particulière sur un périphérique
% TIME	Initialisation de la date
% X	Abandon du background et dump post mortem
% Y	Abandon du background sans dump post mortem
% MODIFY	Modification mémoire
% IT	Armement et/ou validation d'interruptions
% ACTIVATE	Excitation d'une interruption
% PM	Protection, déprotection

4-2. Commande existant sous MTRD mais étendue sous MMT

La commande LOAD qui permet le chargement d'un IMT a été étendue sous MMT pour permettre le chargement d'un programme résident support de tâches différées résidentes.

La syntaxe de la commande LOAD est alors :

$$\%\ \text{LOAD/F,} \begin{Bmatrix} \text{EP,nom} \\ \text{BI} \\ \text{GI} \end{Bmatrix}, [\,@\,[\,\&\,]\,\text{xxxx}\,], \text{Ri}$$

— Le chargement est obligatoirement fait en zone Foreground (paramètre F).

— @ [&] xxxx indique l'adresse relative par rapport au début de la zone Foreground.

— i est le numéro de programme résident (numéro en file d'attente) : O < i < maximum défini à la génération.

Contrôles supplémentaires :

. paramètre R sans paramètre F : % % EC1

. débordement du dictionnaire : % % EC2

4-3. Commande particulière à MMT

La commande TASK permet de créer une tâche :

% TASK/nom/[Pi], [Xj], [Dk], [R]/paramètres/

— i, j, k sont de la forme [&] xxxx

— i est le numéro de priorité (0 par défaut)

— si X est présent la valeur j est chargée dans le registre X

— si X est absent :

. Si la zone paramètres est vide (retour chariot derrière le /) ou ne contient que des blancs, la valeur 0 est chargée dans le registre X.

. Si la zone paramètres contient l caractères (après élimination des blancs à gauche) :
  - un bloc dynamique de taille l est réservé en zone commune dynamique,
  - les l caractères sont placés dans ce bloc,
  - le registre X est chargé avec l'adresse de ce bloc relativement à la zone commune.

— R demande la création de la tâche sur un programme résident (voir cas bit T = 0 pour appel à M : TASK).

— k est la valeur d'un délai en unités échelle 2 (0 par défaut).

Contrôles :

— Nom de tâche inconnu : % % RT01

— Saturation du nombre de tâches : % % RT02

— Priorité illégale : % % RT04

— Partition trop petite : % % RT07.

5. Utilisation des modules moniteur

5-1. Généralités

5-1.1. Classification des modules

On peut distinguer :

- Les modules existant sous MTRD et totalement identiques sous MMT.
- Les modules existant sous MTRD mais étendus sous MMT.
- Les modules particuliers à MMT.

5-1.2. Modules identiques sous MTRD

Leur mode d'utilisation détaillé est précisé dans le manuel d'utilisation de MTRD.

M : HXBN	Conversion hexadécimal - binaire
M : BHHX	Conversion binaire - hexadécimal
M : DCBN	Conversion décimal - binaire
M : BNDC	Conversion binaire - décimal
M : ASEB	Conversion code ASCII - code EBCDIC
M : EBAS	Conversion code EBCDIC - code ASCII
M : LOAD	Chargement d'un IMT en mémoire
M : KEY	Lecture ou écriture clés et voyants pupître
M : DUMP	Dump mémoire sur LO
M : IT	Contrôle inividuel des interruptions
M : CMPA	Comparaison de deux addresses
M : CMPS	Comparaison de deux chaînes de caractères
M : BIB	Recherche en bibliothèque système
M : TYPE	Dialogue sur CB en zone programme
M : WLST	Mise en attente d'une liste d'événements
M : CSOL	Communication avec l'IT pupître
M : TIME	Lecture de l'heure
M : FILE	Déclaration de fichier SGF 15
M : AFF	Lecture de l'affectation d'une étiquette opérationnelle
M : ASGN	Assignation d'une étiquette opérationnelle
M : PM	Modification de la protection mémoire

La plupart de ces modules sont prévus avec en entrée des adresses relatives à G : le verrouillage en mémoire d'une tâche appelante résidente sur disque doit être fait par l'utilisateur si l'adresse transmise n'est pas dans la tâche elle-même.

5-1.3. Modules étendus aux tâches différées

M : EXIT	Fin normale d'une tâche
M : IO	Demande d'entrée/sortie en zone programme
M : ZIO	Demande d'entrée/sortie en zone commune
M : WAIT	Attente d'événement en zone programme
M : ZWAT	Attente d'événement en zone commune
M : ZACT	Activation d'un événement en zone commune
M : ACTV	Activation d'un événement en zone programme
M : ZDLY	Lancement d'un délai en zone commune
M : DLAY	Lancement d'un délai en zone programme
M : SDLY	Suppression d'un délai en zone programme
M : ZSDL	Suppression d'un délai en zone commune

M : RQST	Réservation d'une ressource
M : TAR	Test de l'état d'une ressource et réservation si la ressource est libre
M : RLSE	Libération d'une ressource
M : ABRT	Auto-destruction d'une tâche
M : KILL	Destruction d'une tâche
M : OPEN	Ouverture d'un fichier SGF 15
M : CLOS	Fermeture d'un fichier SGF 15

La plupart de ces modules sont prévus avec en entrée des adresses relatives à G : le verrouillage en mémoire d'une tâche appelante résidente sur disque doit être fait par l'utilisateur si l'adresse transmise n'est pas dans la tâche elle-même (swapping).

Ces modules peuvent avoir des interfaces étendues par rapport au MTRD, mais les interfaces MTRD restent compatibles.

5-1.4. Modules particuliers à MMT

M : TASK	Création d'une tâche
M : QUIT	Appel au scheduler
M : ALOC	Allocation d'un bloc de mémoire dynamique
M : FREE	Libération d'un bloc de mémoire dynamique
M : TAL	Test de disponibilité d'un bloc de mémoire dynamique et réservation si disponible
M : LOCK	Verrouillage d'une tâche en mémoire
M : UNLK	Déverrouillage d'une tâche
M : MAST	Passage d'une tâche en mode maître
M : SLAV	Passage d'une tâche en mode esclave
M : PRIO	Changement de la priorité d'une tâche
M : WDLY	Mise en attente d'un délai pur

5-1.5. Interface tâche—moniteur au lancement de la tâche

Lors du lancement d'une tâche immédiate les registres sont initialisés de la manière suivante pour le moniteur :

A - Bits 0-7	: niveau de la tâche appelante (niveau de l'interruption pupitre en général)
Bits 8-15	: niveau de la tâche lancée
E	: chargé avec 0
X	: paramètre spécifié dans la commande % RUN

Lors du lancement d'une tâche différée les registres sont initialisés de la manière suivante par le moniteur :

A - Bits 0-7	: numéro de la tâche appelante (tâche mère)
Bits 8-15	: numéro de la tâche lancée (tâche fille)
E - Bits 0-7	: priorité de la tâche lancée
Bits 8-15	: 82 : tâche différée non résidente 842 : tâche différée résidente
X	: paramètre spécifié dans le troisième mot du TD.

Lors du lancement d'un programme background les registres sont initialisés de la manière suivante par le moniteur :

A = 0	: lancement sous le moniteur d'enchaînement : BATCH
≠ 0	: lancement sous contrôle de l'interruption pupitre
E = 0	: lancement par les commandes % LOAD - % RUN
≠ 0	: lancement par la commande % CALL
X	: adresse /ZC du tampon où se trouve les paramètres de la commande

5-1.6. Messages d'erreur propres au MMT

En plus des messages d'erreur édités par le MTRD le MMT édite les messages suivants :

%% RT01	Nom de programme inconnu
%% RT02	Saturation du nombre de tâches
%% RT03	Saturation du nombre de tâches à créer sur disque
%% RT04	Priorité illégale
%% RT05	Pas de partition disque disponible
%% RT06	Chargement de tâche différée en mode maître
%% RT07	Partition middleground trop petite
%% RT08	Conflit de swapping Le swapping qui ne peut être réalisé pour une des raisons suivantes, - tâche vérouillée en mémoire - erreur de transfert disque - partition disque non disponible, sera réalisé ultérieurement
%% RT0B	Numéro de ressource illégal
%% RT0F	Adresse de CB invalide.

5-2. Module M : EXIT

5-2.1. Fonction

Fin d'une tâche immédiate ou différée.

5-2.2. Interface

Entrée :

Néant.

Sortie :

- non significatif pour tâche différée,
- pour tâche immédiate : (A) = n° de tâche, (E) = 0.

5-2.3. TWB utilisée

16 mots.

5-2.4. Conditions d'utilisation

Pour l'utilisation répétitive en tant que tâche différée résidente en mémoire ou tâche immédiate, on doit être capable de réinitialiser le programme associé (réinitialisation des données).

5-2.5. Remarques

Après un appel à M : EXIT :

- Une tâche immédiate peut être relancée par M : IT ou % ACTIV.
- Une tâche différée résidente disparaît mais une autre tâche pourra être créée sur le même contexte : cette tâche démarrera sur l'instruction suivant le M : EXIT (c'est là qu'on peut placer la séquence de réinitialisation et le branchement vers le début du programme).
- Une tâche différée non résidente disparaît.
- Un programme tournant en background ne peut pas être relancé.

5-2.6. Actions effectuées

Pour le background :

- Abort des entrées/sorties lancées par le niveau 0.

Pour les tâches différées :

- Libération des blocs de données dynamiques et des ressources possédés par la tâche.
- Libération de la place mémoire et de la zone de swapping pour les tâches résidentes sur disque et libération du programme pour les tâches résidentes en mémoire.

5-3. Module M : IO

5-3.1. Fonction

Demande d'entrée/sortie sur un CB en zone programme.

5-3.2. Interface

Entrée :

(A) = adresse d'un CB relative au G appelant.

Sortie :

(A) = inchangé.
(X) = 0 si demande prise en compte.

5 - 3.3. TWB utilisée

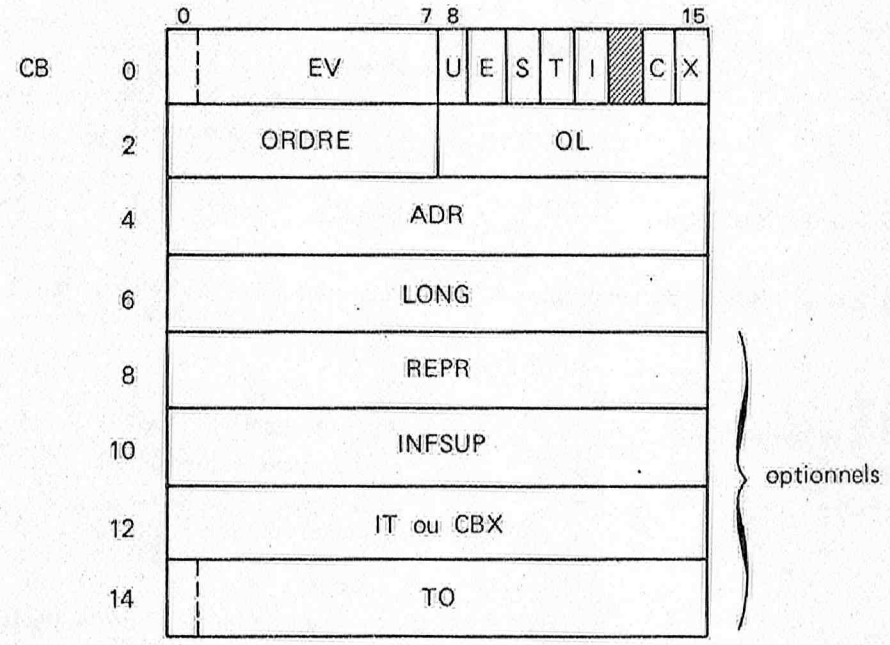
16 mots pour un transfert classique.

32 mots pour SGF 15.

5-3.4. Conditions d'utilisation

- La tâche appelante initialise un transfert par M : IO. A ce transfert est associé un tampon en zone programme. Le verrouillage de la tâche en mémoire pendant tout le temps du transfert est assuré par le système.
- On peut associer un time-out à la demande de transfert.
- La tâche se met en attente de la fin de transfert par un appel à M : WAIT sur le même CB. Cette fin de transfert peut être provoquée par time-out.
- Un CB ne peut pas être réutilisé pendant la durée du transfert : le contrôle est assuré par M : IO.
- La tâche appelante peut exécuter des traitements en simultanéité avant d'appeler M : WAIT.

5-3.5. Control Block (CB) d'entrée/sortie



Mot 1 :

EV — Octet événement.
. Bit 0 = 0 Si événement arrivé. Les bits 1 à 7 contiennent alors le compte rendu.
. Bit 0 = 1 Si événement non arrivé.

U — Contrôle complet des erreurs par l'utilisateur (sauf périphérique non prêt).

E — Adresse de reprise en cas d'erreur (REPR).

S — Présence d'une information supplémentaire (INFSUP).

T — Time-out associé au transfert (TO).

I	Excitation d'IT à la fin du transfert (IT).
C	Contrôle complet de toutes les erreurs.
X	Activation de l'événement CBX associé à la fin de transfert.

Le bit 13 du premier mot est réservé, mais doit être initialisé à la valeur 0.

Remarque :

- Le bit C est prioritaire sur le bit U.
- Le bit I est prioritaire sur le bit X.

Mot 2 :

ORDRE	Commande spécifique au périphérique.
OL	Etiquette opérationnelle.

Mot 3 :

ADR	Adresse du tampon relative à l'appelant.

Mot 4 :

LONG	Compte d'octets à transférer.

Mot 5 :

REPR	Adresse de reprise relative à l'appelant.

Mot 6 :

INFSUP	Information supplémentaire, par exemple adresse secteur pour transfert disque.

Mot 7 :

CBX	Adresse d'un CB d'événement relative à la zone commune : cet événement sera activé en même temps que l'événement associé au CB courant.
IT	Numéro d'interruption à exciter.

Mot 8 :

TO	Valeur d'un délai jouant le rôle de time-out pour le transfert. Si bit 0 = 0 : Valeur exprimée en échelle 1 (100 ms). Si bit 0 = 1 : Valeur exprimée en échelle 2 (10s).

5-3.6. Contrôle des transferts

On distingue :

- les erreurs récupérables (% EOD par exemple),
- les erreurs irrécupérables.

Dans le premier cas le contrôle est toujours rendu sur le WAIT avec un compte rendu (bit 1 = 0).

Dans le deuxième cas :

- Si le bit C = 1 le contrôle est toujours rendu à l'utilisateur avec un compte rendu.
- Si le bit C = 0 :
  - Si bit U = 0 : édition d'un message sur OC et abort de la tâche appelante.
  - Si bit U = 1 : le contrôle est rendu à l'utilisateur avec un compte rendu.

Dans le cas de reprise périphérique le système est toujours en attente d'une remise en service quelque soit la valeur des bits C et U.

5-3.7. Messages d'erreur

Ces messages sont édités sur OC.

%%IO00	Time-out sur périphérique ne possédant pas d'ordre STOP.
%%IO01	Erreur logique en fin de transfert.
%%IO02	Erreur logique en début de transfert.
%%IO03	Erreur physique en fin de transfert.
%%IO04	Erreur physique en début de transfert.

5-4. Module M : ZIO

5-4.1. Fonction

Demande d'entrée/sortie sur un CB en zone commune.

5-4.2. Interface

Entrée :

(A) = adresse d'un CB relative à la zone commune.

Sortie :

(A) = inchangé.

X = 0 si demande prise en compte.

X ≠ 0 si demande non prise en compte.

5-4.3. TWB utilisée

16 mots pour une demande de transfert classique.

32 mots pour SGF 15.

5-4.4. Conditions d'utilisation

- Mêmes remarques que pour l'appel à M : IO.
- Le tampon associé au transfert est lui aussi en zone commune.
- Le transfert ayant lieu en zone commune, la tâche appelante peut être swappée-out pendant le transfert.
- La tâche en attente de la fin de transfert (qui peut être différente de la tâche appelante) peut également être swappée-out pendant le transfert.
- La structure du CB et le contrôle des transferts sont identiques à ceux de M : IO.

5-5. Module M : WAIT

5-5.1. Fonction

Attente d'arrivée d'un événement sur un control block (CB) en zone programme.

5-5.2. Interface

Entrée :

(A) = adresse d'un CB relative au G appelant.

Sortie :

(A) < 0 Si erreur et bit E = 0 dans CB ou délai arrivé le premier.

(A) ⩾ 0 Si pas d'erreur ou si erreur et bit E = 1 dans le CB (A contient alors l'adresse par rapport à G du CB de l'événement).

5-5.3. TWB utilisée

16 mots.

5-5.4. Conditions d'utilisation

- Pour que l'attente soit effective il faut que le bit 0 de l'octet événement ait été positionné à 1 (sauf cas &A0 qui rend les WAIT ineffectifs mais pas les ACTV) :
  - par l'utilisateur lui-même,
  - par un appel à M : IO (E/S),
  - par un appel à M : DLAY (délai),
- Une attente est débloquée par un appel à M : ACTV qui transmet un compte rendu dans l'octet événement.
- Dans le cas particulier d'un événement ou délai, le compte rendu permet de distinguer les 2 cas :
  - (A) < 0 et compte rendu = &4E, le délai est arrivé le premier.
  - (A) < 0 et compte rendu ≠ &4E, l'événement est arrivé le premier et il y a erreur.

- Le cas A ⩾ 0 correspond toujours au cas sans erreur.
- L'adresse du CB étant relative à G, la tâche appelante ne peut pas être swappée-out pendant l'attente : le verrouillage en mémoire est automatiquement fait par le système.

5-5.5. Messages d'erreur

Une tentative de mise en attente sur un événement qui aurait été détruit se traduit de la façon suivante :

- Si l'octet détruit est positif, l'appelant n'est pas mis en attente de l'événement. Les tâches qui pouvaient déjà être en attente avant la destruction ne seront jamais relancées.
- Si l'octet détruit est négatif :
  . le message % % RTOF est édité sur OC.
  . le contrôle est rendu à l'utilisateur avec (A) < 0 et octet événement nul.

5-5.6. Partie d'un CB concernée par le module M : WAIT

Quelque soit le type d'événement (entrée/sortie, entrée/sortie + time-out, événement utilisateur) M : WAIT utilise les éléments suivants :

Mot 1 du CB

. Bit 0 = 1 si événement non arrivé,
  Bit 0 = 0 si événement arrivé.
. Si événement arrivé bits 1 à 7 = compte rendu ; bit 1 = 1 signifie erreur irrécupérable et le bit 9 est examiné.

Mot 5 du CB si le bit 9 du mot 0 a la valeur 1 (E = 1) et qu'il y a eu détection d'une erreur irrécupérable : M : WAIT effectue alors un branchement à l'adresse contenue dans ce mot.

5-6. Module M : ZWAT

5-6.1. Fonction

Attente de l'arrivée d'un événement sur un control block (CB) en zone commune.

5-6.2. Interface

Entrée :
(A) = adresse d'un CB relative à la zone commune.

Sortie :
Comme pour M : WAIT.

5-6.3. TWB utilisée

16 mots.

5-6.4. Conditions d'utilisation

- Mêmes remarques que pour M : WAIT.
- Activation par M : ZACT.
- L'événement étant matérialisé par un CB en zone commune, la tâche en attente peut être swappée-out pendant l'attente.

5-7. Module M : ZACT

5-7.1. Fonction

Activation d'un événement matérialisé par un control block (CB) en zone commune.

5-7.2. Interface

Entrée :
(A) = adresse d'un CB relative à la zone commune.
(E) bits 9-15 = compte rendu.

Sortie :
(A) < 0 = CB incorrect.
(A) ⩾ 0 = Activation effective.

5-7.3. TWB utilisée

16 mots

5-7.4. Conditions d'utilisation

- Permet de relancer toutes les tâches en attente de l'événement.
- Ces tâches peuvent avoir été swappées-out pendant l'attente.
- Pour les tâches immédiates la relance consiste en une activation d'interruption effectuée par le niveau qui effectue M : ZACT.
- Pour les tâches différées la relance consiste en une remise en l'état éligible effectuée au niveau 1 avant l'exécution du prochain appel au Scheduler.
- Avant la relance des tâches en attente, le compte rendu est recopié dans les bits 1 à 7 du CB.
- Le bit 0 du CB est remis à 0 (événement arrivé).
- L'activation d'un événement déjà arrivé est ineffective.

5-7.5. Contrôles

Une tentative d'activation d'un événement qui aurait été détruit se traduit de la façon suivante :

- Si l'octet détruit est positif : activation ineffective,
- Si l'octet détruit est négatif :
  . message %%RTOF édité sur OC,
  . contrôle rendu à l'utilisateur avec (A) < 0 et compte rendu nul.

5-7.6. IT auxiliaire

Si le bit I = 1 dans le CB, l'activation se traduira en plus par l'excitation du numéro d'IT contenu dans le septième mot du CB.

Cette activation a lieu avant la relance des tâches en attente.

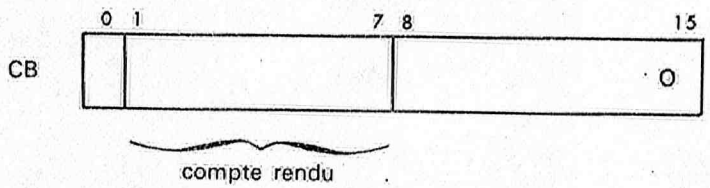
5-7.7. Evénement auxiliaire

Si le bit X = 1 dans le CB, l'activation se traduira en plus par l'activation de l'événement dont l'adresse relative à la zone commune est contenue dans le septième mot du CB.

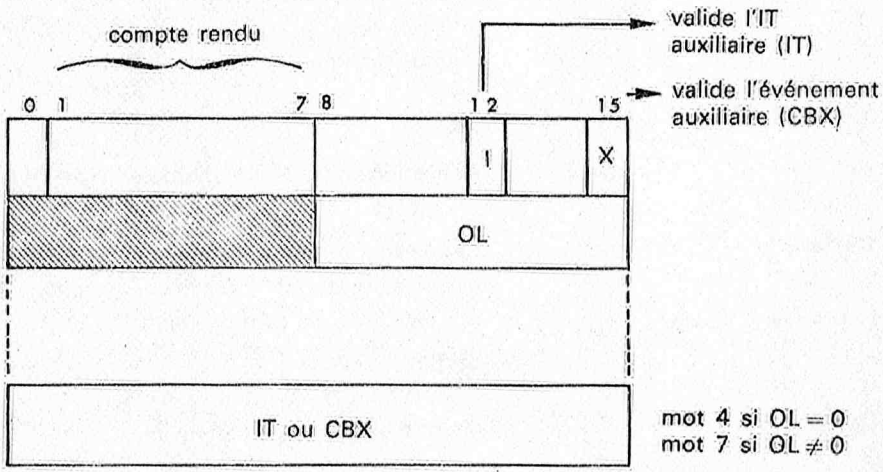
Cette activation a lieu après l'activation primaire.

5-7.8. Partie d'un CB concernée par le module M : ZACT

a) Evénement simple :



b) Evénement associé à un délai ou time-out :



Les bits 0 à 7 du deuxième mot sont réservés mais doivent être initialisés à la valeur 0.

OL = 0 pour délai simple

OL = Etiquette opérationnelle (≠ 0) pour time-out.

Bit 0 du mot 1 :

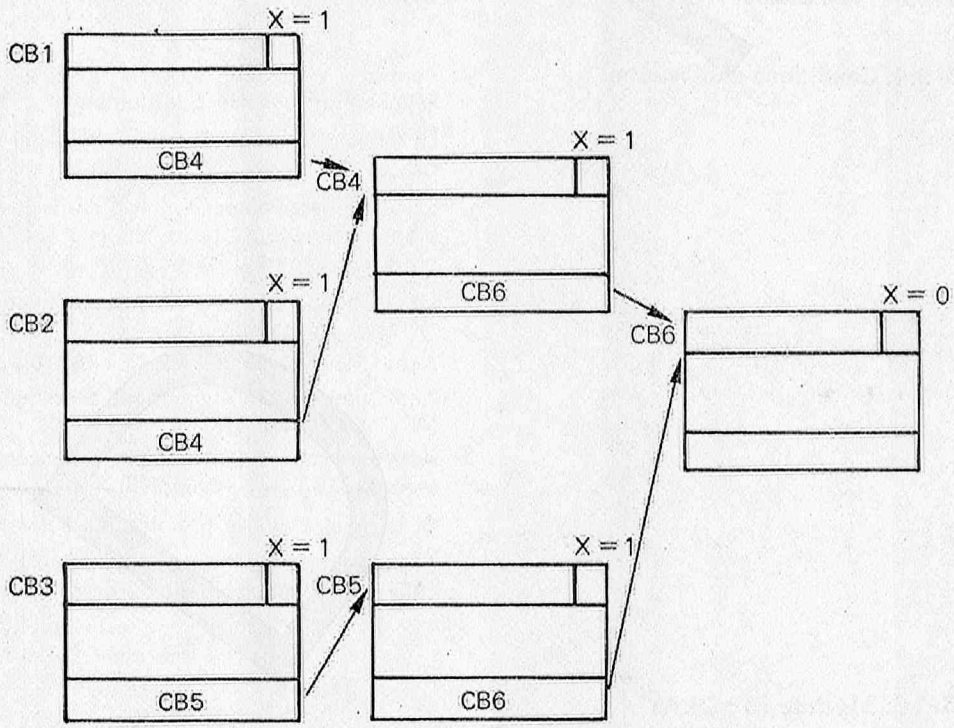
. 0 si événement arrivé.
. 1 Si événement non arrivé.

Le bit I est prioritaire sur le bit X :

. Si I = 1, IT contient un numéro d'interruption à activer.
. Si I = 0 et X = 1, CBX contient l'adresse relative à la zone commune d'un événement qui sera activé.

5-7.9. Chaînage d'événements

Dans cet exemple l'événement associé à CB6 sera activé par l'activation de l'un quelconque des événements CB1 à CB6.



5-8. Module M : ACTV

5-8.1. Fonction

Activation d'un événement matérialisé par un control block (CB) en zone programme.

5-8.1. Interface

Entrée :

(A) = Adresse d'un CB relative au G appelant.

(E) Bits 9-15 = compte rendu.

Sortie :

(A) < 0 = CB incorrect.

(A) ⩾ 0 = Activation effective.

5-8.3. TWB utilisée

16 mots.

5-8.4. Conditions d'utilisation

— Mêmes remarques que pour M : ZACT.

— L'événement étant en zone programme, et, activation et mise en attente ne pouvant avoir lieu dans la même tâche, cela implique :

. que l'appelant soit bloqué en mémoire ou résident.
. qu'il puisse accéder au CB de l'autre tâche.

— En cas de présence du bit X il y aura activation d'un événement auxiliaire qui sera référencé par une adresse relative au début de la zone commune.

5-9. Module M : ZDLY

5-9.1. Fonction

Lancement d'un délai matérialisé par un control block (CB) en zone commune.

5-9.2. Interface

Entrée :

(A) = Adresse d'un CB relative à la zone commune.

Sortie :

(A) = Inchangé

(X) = 0 Délai pris en compte.

(X) ≠ 0 Délai non pris en compte.

5-9.3. TWB utilisé

16 mots.

5-9.4. Conditions d'utilisation

- Permet à une tâche de créer un délai dont l'arrivée se traduira par l'activation de l'événement associé à la demande.
- La tâche se met en attente du délai par un appel au module M : ZWAT sur le même CB.
- La tâche sera relancée, soit à l'arrivée du délai, soit par une activation de l'événement par une autre tâche (appel à M : ZACT) : ceci permet de réaliser la fonction attente d'événement ou de délai.
- La distinction entre les 2 types d'activation est effectuée par le module M : ZWAT de la façon suivante :
  - Activation due au délai : (A) < 0, CB_{0-7} = &4E
  - Activation due à l'événement avec compte rendu de non erreur : (A) ⩾ 0, CB_{0-7} < &40
  - Activation due à l'événement avec compte rendu d'erreur : (A) < 0, CB_{0-7} ⩾ &40 (≠&4E)
- Si le délai arrive le premier, les activations ultérieures sont ineffectives.
- Si l'activation de l'événement arrive la première, les autres activations, y compris celle due à l'épuisement du délai, sont ineffectives.

5-10. Module M : DLAY

5-10.1. Fonction

Lancement d'un délai matérialisé par un CB en zone programme.

5-10.2. Interface

Entrée :

(A) = Adresse d'un CB relative au G appelant.

Sortie :

(A) = Inchangé.

(X) = 0 Délai pris en compte.

(X) ≠ 0 Délai non pris en compte.

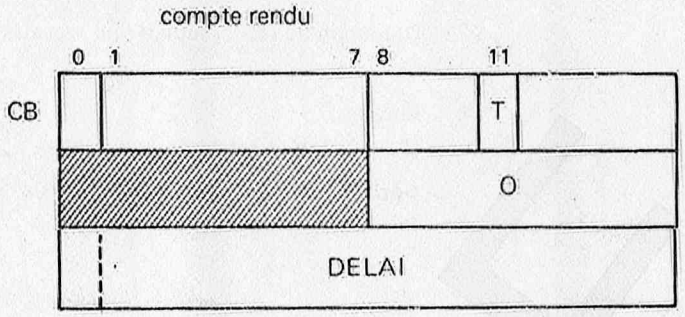
5-10.3. TWB utilisée

16 mots.

5-10.4. Conditions d'utilisation

- Mêmes remarques que pour M : ZDLY.
- Le CB étant en zone programme, il ne pourra être réactivé que dans la mesure où il reste implanté de façon fixe en mémoire : la tâche appelante doit être verrouillée en mémoire ou résidente.
- L'activation pourra également être faite par appel au module M : ACTV.
- L'arrêt du délai sera effectué par un M : SDLY si nécessaire.

5-10.5. Partie d'un CB concernée par le module M : DLAY



Les bits 0 à 7 du deuxième mot sont réservés mais doivent être initialisés à la valeur 0.

Le bit 0 du mot signifie :
- 0 événement arrivé,
- 1 événement non arrivé.

Le bit 11 doit être positionné à 1 (T) : il sert à valider la valeur du délai qui est indiqué par DELAI en échelle 1 ou 2 :

DELAI { . Bit 0 = 0 échelle 1
. Bit 0 = 1 échelle 2.

5-11. Module M : SDLY

5-11.1. Fonction

Suppression d'un délai matérialisé par un CB en zone programme.

5-11.2. Interface

Entrée :
(A) = Adresse d'un CB relative au G appelant.

Sortie :
(A) = 0 Délai supprimé.
(A) < 0 Demande incorrecte, il n'existe pas de délai associé à ce CB.

5-11.3. TWB utilisée

16 mots.

5-11.4. Conditions d'utilisation

Une demande de délai sur un CB sera supprimée si elle existe. Dès qu'elle sera supprimée, il n'y aura plus d'activation de l'événement associé.

5-12. Module M : ZSDL

5-12.1. Fonction

Suppression d'un délai matérialisé par un CB en zone commune.

5-12.2. Interface

Entrée :
(A) = Adresse d'un CB relative à la zone commune.

Sortie :
(A) = 0 Délai supprimé.
(A) < 0 Demande incorrecte, il n'existe pas de délai associé à ce CB.

5-12.3. TWB utilisée

16 mots.

5-12.4. Conditions d'utilisation

Mêmes remarques que pour M : SDLY.

5-13. Module M : RQST

5-13.1. Fonction

Demande de réservation d'une ressource.

5-13.2. Interface

Entrée :
(A) = Numéro de la ressource.

Sortie :
Pas de sens.

5-13.3. TWB utilisée

16 mots.

5-13.4. Conditions d'utilisation

- La ressource est un outil de protection.
- La réservation d'une ressource transmet la propriété de la ressource à la tâche appelante.
- La demande est ineffective si la tâche appelante possédait déjà la ressource.
- Si la ressource n'est pas libre, la tâche est mise en file d'attente.
- Les ressources possédées par une tâche sont libérées si la tâche est abortée.
- La mise en attente est faite de la façon suivante :
  . tâche immédiate : elle est mise dans la file à un rang qui correspond à son niveau,
  . tâche différée : elle est mise à la fin de la file d'attente.
- Certaines ressources sont réservées au système (en standard 0 à 9).

5-13.5. Erreur

Une demande de réservation avec un numéro illégal se traduit par l'édition d'un message %%RTOB sur ,OC, et la tâche appelante est détruite (abort).

5-14. Module M : TAR

5-14.1. Fonction

Test de l'état d'une ressource et réservation si la ressource est libre.

5-14.2. Interface

Entrée :
(A) = Numéro de la ressource.

Sortie :
(A) $<$ 0 Si ressource occupée.
(A) $\geqslant$ 0 Si ressource réservée.

5-14.3. TWB utilisée

16 mots.

5-14.4. Conditions d'utilisation

- Mêmes remarques que pour le module M : RQST.
- Même traitement si la ressource est occupable sans attente.
- Contrôle rendu avec (A) < 0 si la demande avait entraîné une mise en attente (la demande est alors abandonnée).

5-14.5. Erreur

Si la demande est faite avec un numéro illégal, la tâche appelante est abortée et il y a édition d'un message % % RTOB sur OC.

5-15. Module M : RLSE

5-15.1. Fonction

Libération d'une ressource.

5-15.2. Interface

Entrée :
(A) = Numéro de la ressource.

Sortie :
Pas de sens.

5-15. 3. TWB utilisée

16 mots.

5-15.4. Conditions d'utilisation

- La libération d'une ressource non réservée est ineffective.
- La ressource ne devient effectivement libre que si la file d'attente associée est vide.

  Sinon la première tâche de la file d'attente est relancée en prenant possession de la ressource.
- Relance signifie : rendre éligible une tâche différée ou exciter le niveau d'une tâche immédiate.
- Si le numéro de la ressource n'existe pas, message % % RTOB sur OC et A < 0.

5-16. Module M : ABRT

5-16.1. Fonction

Auto-destruction d'une tâche immédiate ou différée et de sa filiation.

5-16.2. Interface

Néant.

5-16.3. TWB utilisée

38 mots.

5-16.4. Conditions d'utilisation

- Identique à M : EXIT
- Réalise en plus :
  . l'abort des transferts en cours pour cette tâche,
  . l'abort des tâches de la filiation.
- Dans le cas d'une tâche non résidente le contexte occupé est perdu par le système.
- Dans le cas d'une tâche résidente, le contexte peut être réutilisé : on peut placer une séquence de réinitialisation derrière l'appel au module M : ABRT.

5-17. Module M : TASK

Ce module est particulier à MMT.

5-17.1. Fonction

Création d'une tâche.

5-17.2. Interface

Entrée :

(A) = Adresse relative à G d'un descripteur de tâche (TD).

Sortie :

(A) = Numéro de la tâche créée.

(A) < 0 Si tâche non créée.

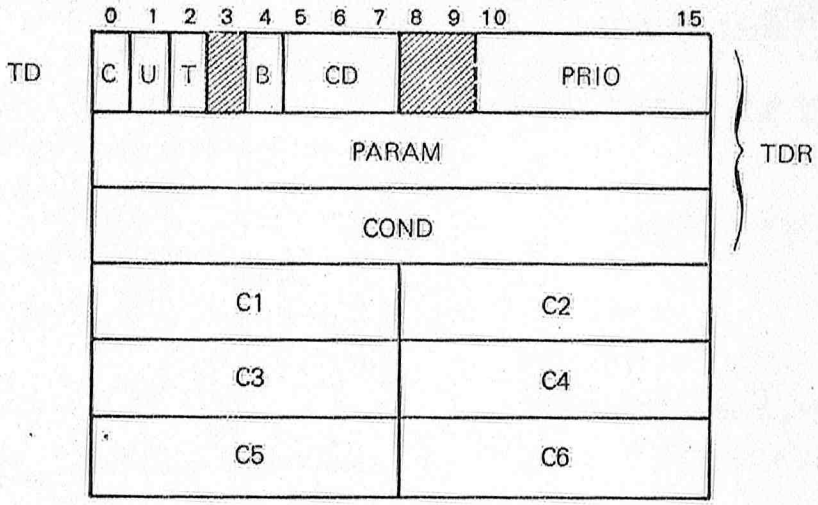
5-17.3. TWB utilisée

32 mots.

5-17.4. Conditions d'utilisation

- La création d'une tâche consiste principalement à associer un numéro de tâche à un programme, c'est-à-dire à prendre une entrée dans les tables de tâches connues du système.
- La création de la tâche entraîne son réveil (éventuellement retardé), c'est-à-dire de sa mise en éligibilité.
- Le nom du programme ainsi que les informations nécessitées par le réveil figurent dans le TD dont une partie est un TDR (TD pour réveil).

5-17.5. Descripteur de création de tâche (TD)



Mot 1 :

Bits C et U	Non traités.
Bit T	Création sur une copie fraiche du programme.
Bit B	Transmission de la possession d'un bloc dynamique.
CD	Spécifie les conditions de lancement (voir 5-17.9.).
PRIO	Priorité de la tâche créée. Les bits 3, 8 et 9 du premier mot sont réservés mais doivent être initialisés à la valeur 0.

Mot 2 :

PARAM	Valeur d'un paramètre ou adresse relative à la zone commune d'un bloc de paramètres.

Mot 3 :

COND	Sa signification est déterminée par CD : — ignoré, — adresse d'un CB en zone programme, — adresse d'un CB en zone commune, — délai.

Mots 4, 5, 6 :

Nom du programme sur lequel est créée la tâche (complété par des blancs à droite).

A la création de la tâche ces trois mots sont modifiés par le système .

5-17.6. Programme associé à la tâche

— Le programme associé à la tâche est défini par son nom.

— A partir du nom, le système obtient les informations localisant le programme grâce :

. au dictionnaire des noms de tâches (en mémoire),

. au catalogue de la bibliothèque EP (sur disque).

— La présence d'un nom dans le dictionnaire est obligatoire pour une tâche résidente et facultative pour une tâche non résidente : dans ce dernier cas une amélioration des performances est apportée par la présence du nom dans le dictionnaire (gain d'un appel disque).

— Dans le cas d'un programme résident, le dictionnaire associe au nom un numéro de programme résident et une file d'attente des tâches en attente de ce programme.

— La recherche du nom transmis dans le TD s'effectue par un balayage du dictionnaire en mémoire éventuellement suivi d'un balayage du catalogue de EP.

. En cas d'échec le contrôle est rendu à la tâche appelante avec (A) < 0 et le message %% RT01 est édité sur OC.

. En cas de succès les informations sont stockées dans le TD à la place du nom pour éviter une nouvelle recherche en cas de réutilisation du TD pour une nouvelle création.

5-17.7. Numéro de tâche

— Le numéro affecté à une tâche lors de sa création, servira à l'identifier par la suite.

— Ce numéro redevient disponible après l'EXIT ou l'abort de la tâche.

— Il peut y avoir saturation de tâches :

. Saturation en mémoire (pas de numéro libre) : le contrôle est rendu à l'utilisateur avec (A) < 0 et le message %% RT02 est édité sur OC,

. Saturation de tâche à créer sur disque : le contrôle est rendu à l'utilisateur avec (A) < 0 et le message %% RT03 est édité sur OC.

Le deuxième cas est normalement évité par un bonne configuration du système.

5-17.8. Création de tâche sur copie fraiche (bit T = 1)

Cette notion a un sens pour des tâches résidentes en mémoire.

— Bit T = 0 :

La tâche est associée à la première occurrence du nom du programme dans le dictionnaire. Si le programme est déjà occupé, la tâche sera mise en file d'attente.

— Bit T = 1 :

On veut éviter la mise en attente de disponibilité du programme :

. Si le nom du programme se trouve plusieurs fois dans le dictionnaire, on associe la tâche au premier programme dont la file d'attente est vide ; en cas d'échec on crée la tâche comme tâche différée non résidente en mémoire.

. Dans le cas T = 1 les informations de localisation du programme ne sont pas recopiées dans le TD (puisque la copie du programme utilisée dépend de l'état du système à la création).

5-17.9. Réveil d'une tâche

— Le réveil d'une tâche consiste en sa mise en éligibilité à la priorité spécifiée.

— Cette mise en éligibilité peut être retardée jusqu'à la réalisation d'une condition spécifiée par certains paramètres du TDR :

COND	Sa signification est déterminée par CD.	
CD	000	Réveil immédiat, COND ignoré.
	001	Réveil sur arrivée d'un événement en zone commune, COND = adresse d'un CB relative à la zone commune.
	010	Réveil sur arrivée d'un événement en zone programme, COND = adresse d'un CB relative à G.
	100	Réveil sur arrivée d'un délai exprimé en échelle 1 ou 2.

— La valeur contenue dans PARAM se retrouvera dans le registre X lors de la relance de la tâche créée.

— Si le TD est localisé en zone commune :

. l'adresse du TD transmise à l'appel doit être relative à G,

. la tâche appelante doit être verrouillée en mémoire.

5-17.10. Erreurs

La tâche appelante est abortée dans tous les cas et un message est édité sur OC :

%%RT01	nom de programme non trouvé.
%%RT02	saturation du nombre de tâches.
%%RT03	saturation tâches à créer sur disque.
%%RT04	priorité illégale.
%%RT05	pas de partition disponible.
%%RT0C	transmission d'un bloc dynamique non possédé par la tâche appelante.
%%RT0F	adresse de CB non valide.

5-18. Module M : QUIT

5-18.1. Fonction

Rendre la main au Scheduler pour une tâche différée.

5-18.2. Interface

Néant.

5-18.3. TWB utilisée

16 mots.

5-18.4. Conditions d'utilisation

— Appel ineffectif pour une tâche immédiate ou le background.

— La tâche appelante perd son rang dans la file d'attente des tâches éligibles associée à sa priorité.

— Cette fonction permet d'éviter que des longs traitements sans attente bloquent les autres tâches de même priorité.

5-19. Module M : ALOC

5-19.1. Fonction

Demande d'allocation d'un bloc dynamique de mémoire.

5-19.2. Interface

Entrée :

(A) = Taille demandée en octets.

Sortie :

(A) = Taille obtenue en octets.

(X) = Adresse relative à la zone commune du bloc obtenu.

(A) < 0 Bloc trop grand.

5-19.3. TWB utilisée

16 mots.

5-19.4. Conditions d'utilisation

— Si un bloc standard de la taille immédiatement supérieure ou égale à la taille demandée est libre, le contrôle est rendu à l'utilisateur après prise de possession du bloc par la tâche appelante.

— Dans le cas contraire le système tente d'obtenir un bloc standard par partitionnement de bloc standard de taille supérieure : si c'est possible, même comportement que précédemment.

— Si cela n'est pas possible, la tâche appelante est mise en attente de la libération d'un bloc de taille suffisante :

La mise en attente s'effectue :

. selon la priorité du niveau pour une tâche immédiate,

. en bout de file pour une tâche différée.

5-20. Module M : FREE

5-20.1. Fonction

Libération d'un bloc dynamique de mémoire.

5-20.2. Interface

Entrée :

(X) = Adresse du bloc à libérer relative à la zone commune.

Sortie :

Néant.

5-20.3. TWB utilisée

16 mots.

5-20.4. Conditions d'utilisation

— Le bloc libéré est rendu au système qui peut éventuellement le reconcaténer avec d'autres blocs libres.

— Une tâche ne peut libérer qu'un bloc qu'elle possède : si ce n'est pas le cas, la tâche appelante édite un message % % RTOC sur OC.

5-20.5. Relance des tâches en attente de bloc

— Le système recherche s'il existe des tâches d'un bloc de la taille reconstituée par la libération.

— Si c'est le cas on relance la première des tâches en attente en lui donnant la possession du bloc.

— Si ce n'est pas le cas, le système recherche s'il y a des tâches en attente pour des blocs plus petits (recherche par taille décroissante) : il peut donc y avoir répartitionnement du bloc.

— Si aucune tâche n'est en attente d'un bloc de taille inférieure, le bloc reconstitué est libéré.

5-21. Module M : TAL

5-21.1. Fonction

Test de la disponibilité d'un bloc dynamique de mémoire de taille donnée et allocation si disponible.

5-21.2. Interface

Entrée :

(A) = Taille demandée en octets

Sortie :

(A) = Taille du bloc obtenu en octets.

(X) = Adresse du bloc obtenu relative à la zone commune.

(A) < 0 Si bloc non obtenu ou trop grand.

5-21.3. TWB utilisée

16 mots.

5-21.4. Conditions d'utilisation

— S'il n'existe aucun bloc standard disponible de taille supérieure ou égale à la taille demandée, le contrôle est rendu à l'utilisateur avec (A) < 0.

— Sinon le premier bloc disponible est partitionné pour obtenir le plus petit bloc standard correspondant à la taille demandée. L'allocation est alors effectuée.

5-22. Module M : LOCK

5-22.1. Fonction

Verrouillage d'une tâche en mémoire.

5-22.2. Interface

Entrée :

(A) = Numéro de la tâche à verrouiller.

(A) < 0 Verrouillage de la tâche appelante.

Sortie :

(A) < 0 Si débordement du compteur ou si tâche immédiate ou illégale.

(A) ⩾ 0 Dans les autres cas.

5-22.3. TWB utilisée

16 mots.

5-22.4. Conditions d'utilisation

- A toute tâche différée est associé un compteur.
- Le module LOCK incrémente ce compteur.
- Tant que le compteur est différent, la tâche est verrouillée en mémoire (elle ne peut pas être swappée-out) : elle reste donc résidente en mémoire à la même adresse.
- La limite du compteur est 31 : en cas de menace de débordement, le contrôle est rendu à l'utilisateur avec (A) < 0.
- Si le numéro de tâche est illégal ou si la tâche est une tâche immédiate, le contrôle est rendu à l'utilisateur avec (A) < 0.

5-23. Module M : UNLK

5-23.1. Fonction

Déverrouillage mémoire d'une tâche.

5-23.2. Interface

Entrée :

(A) = Numéro de la tâche à déverrouiller.

(A) < 0 Déverrouillage de la tâche appelante.

Sortie :

(A) < 0 Si compteur nul ou tâche immédiate.

(A) ⩾ 0 Dans les autres cas.

5-23.3. TWB utilisée

16 mots.

5-23.4. Conditions d'utilisation

- Le module UNLK décrémente le compteur de verrouillage.
- La tâche ne sera effectivement déverrouillée (c'est-à-dire swappable-out) que si le compteur est redevenu nul.
- Si le compteur est déjà nul, l'appel est ineffectif et le contrôle est rendu à l'utilisateur avec (A) < 0. Il en est de même si la tâche à déverrouiller est une tâche immédiate.
- Certains modules systèmes peuvent locker la tâche appelante en mémoire (c'est le cas de M : IO). Ces modules déverrouilleront toujours la tâche.

5-24. Module M : MAST

5-24.1. Fonction

Passage de l'appelant en mode maître.

5-24.2. Interface

Entrée :

Néant.

Sortie :

(A) < 0 Si débordement du compteur de verrouillage ou appelant tâche immédiate.

(A) ⩾ 0 Dans les autres cas.

5-24.3. TWB utilisée

16 mots.

5-24.4. Conditions d'utilisation

- Le contrôle est rendu à l'appelant avec passage en mode maître, mais l'accès aux zones protégées reste ce qu'il était avant l'appel.
- La tâche appelante pourra donc exécuter les instructions privilégiées mais ne pourra plus utiliser les modes d'adressage IL et ILX à moins d'avoir prévu des additions du contenu de G aux pointeurs.
- Le passage en mode maître permet cela en s'accompagnant d'un verrouillage implicite en mémoire (LOCK) : en cas de menace de débordement du compteur de verrouillage, le contrôle sera rendu à l'utilisateur avec (A) < 0, mais le passage en mode maître aura eu lieu ; il en est de même si la tâche appelante est une tâche immédiate.
- Une tâche différée non résidente est toujours à l'origine en mode esclave.
- Une tâche différée résidente peut être chargée en mode maître.
- Il faut respecter l'équilibre des appels à MAST et à SLAV (problème du verrouillage).
- Un appel à MAST quand on est déjà en mode maître est équivalent à un appel à LOCK.

5-25. Module M : SLAV

5-25.1. Fonction

Passage de l'appelant en mode esclave.

5-25.2. Interface

Entrée :

Néant.

Sortie :

(A) < 0 Si compteur de verrouillage nul ou appelant tâche immédiate.

(A) ⩾ 0 Dans les autres cas.

5-25.3. TWB utilisée

16 mots.

5-25.4. Conditions d'utilisation

- Le contrôle est rendu à l'appelant avec passage en mode esclave.
- Le mode de protection n'est pas modifié.
- Un déverrouillage mémoire implicite est effectué sur la tâche appelante : si le compteur de verrouillage était nul ou si la tâche appelante était une tâche immédiate, le contrôle est rendu à l'utilisateur avec (A) < 0.
- Un appel à SLAV quand on est en mode esclave est équivalent à un UNLK.

5-26. Module M : PRIO

5-26.1. Fonction

Changement de priorité d'une tâche différée.

5-26.2. Interface

Entrée :

(A) = Numéro de la tâche concernée.

(E) = Valeur de la nouvelle priorité.

Sortie :

(A) < 0 Si la tâche concernée est une tâche immédiate, ou si numéro illégal, ou si tâche non créée, ou si priorité illégale.

(A) ⩾ 0 Dans les autres cas.

5-26.3. TWB utilisée

16 mots.

5-26.4. Conditions d'utilisation

- La tâche concernée peut être la tâche appelante.
- Si la tâche spécifiée est en attente ou dormante, sa priorité est substituée sans autre conséquence.

- Si la tâche spécifiée est éligible, elle est retirée de la file d'attente de l'ancienne priorité pour être mise dans celle de la nouvelle priorité, perdant ainsi son rang.
- Si la tâche spécifiée est la tâche appelante elle ne perd pas le contrôle car le Scheduler n'est pas appelé par le module PRIO.
- Si la priorité spécifiée est identique à l'ancienne priorité, la tâche perd son rang dans la file d'attente d'éligibilité associée à cette priorité.

5-27. Module M : WDLY

5-27.1. Fonction

Attente de l'arrivée d'un délai pur.

5-27.2. Interface

Entrée :

(A) = Valeur du délai (en échelle 1 ou 2).

Sortie :

(A) $\geqslant$ 0 Délai mis en compte.

(A) $<$ 0 Délai non pris en compte.

5-27.3. TWB utilisée

16 mots.

5-27.4. Conditions d'utilisation

- Ce module réalise simultanément les fonctions de création d'un délai et de mise en attente de la tâche appelante.
- L'arrivée du délai n'est matérialisée par aucune structure en zone programme ou zone commune. Il s'agit d'un "délai pur", dont la valeur est exprimée en échelle 1 (bit 0 = 0) ou échelle 2 (bit 0 = 1).
- La tâche en attente pourra donc être swappée-out durant l'écoulement du délai.

cii COMPAGNIE INTERNATIONALE POUR L'INFORMATIQUE

RC : 669805764 B
R.C. Versailles – SIRENE : 669805764

Siège Social
Direction Commerciale
Division des Petits Ordinateurs
et des Applications Spécialisées
Direction Après-Vente
68, Route de Versailles
78430 Louveciennes
Tél. 954 9080

Direction Générale
Institut de Formation
Parc de Rocquencourt
78150 Le Chesnay
Tél. 954 4400

Centre de Vélizy
Division Militaire Spatiale
et Aéronautique
Direction Après-Vente
10 - 12 avenue de l'Europe
78140 Vélizy
Tél. 946 9670

Centre des Clayes-sous-bois
Avenue Jean Jaurès
78340 Les Clayes-sous-bois
Tél. 055 8000

Centre de Toulouse
Avenue du Général Eisenhower
31023 Toulouse
Tél. (61) 40 1140.

DÉLÉGATIONS RÉGIONALES

RHONE-ALPES
177, rue Garibaldi Immeuble M + M
69003 Lyon
Tél. (78) 62 9065

Tour Mont Blanc
15, bd. du Maréchal Leclerc
38000 Grenoble
Tél. (76) 44 9922

18-20 av. du Maréchal Foch
21000 Dijon
Tél. (80) 32 2047

OUEST
3 Place du Colombier
35000 Rennes
Tél. (99) 30 8454

CENTRE-OUEST
9, place Rouget de Lisle
37000 Tours
Tél. (47) 20 2209

MIDI-PYRÉNÉES
Av. du Général Eisenhower
31023 Toulouse
Tél. (61) 40 3563

SUD-EST
433, rue Paradis
13008 Marseille
Tél. (91) 77 0994

AQUITAINE
353, bd du Président Wilson
33200 Bordeaux
Tél. (56) 08 6363

EST
25, avenue Robert Schuma
57000 Metz
Tél. (87) 68 4921

15, rue des Francs Bourgeois
67000 Strasbourg
Tél. (88) 32 1103

NORD
13, boulevard de la Liberté
59000 Lille
Tél. (20) 57 7353