

**LABORATOIRE D'INFORMATIQUE THEORIQUE  
& APPLICATIONS DE MARSEILLE  
L.I.T.A.M.**

Faculté des sciences Economiques

**UNIVERSITE d'AIX-MARSEILLE II**

---

**ISSN 0291 - 5413**

<b>INFORMATIQUE FONDAMENTALE &amp; APPLICATIONS</b>
<b>Comité de rédaction</b>
<b>E. Bianco R. Cusin P. Isoardi J.P. Lehmann R. Stutzmann</b>
<b>Dépositaire B.U. Sc. Eco. Aix-Mars. II</b>

**BULLETIN 28**

**SOMMAIRE**

**P1 ... EDITORIAL:**

Informatique et Europe.  
E. Bianco

**P5 ... Système de Conduite de processus.**

S. Hilala.

**P40 ... Douzzauedibisar**

**Mars 1991**

**Adresse postale : FACULTÉ DES SCIENCES ECONOMIQUES  
LITAM**

**14 rue Puvis de CHAVANNES 13001 MARSEILLE  
91 90 13 20 P 420 et 421**



## EDITORIAL

...Ainsi on arrive à l'intelligence aveugle....  
Tandis que les média produisent la basse  
crétinisation, l'Université produit la haute  
crétinisation. ...

E. Morin : Introduction à la pensée complexe.

### INFORMATIQUE ET CRETINISME.

Imaginons simplement qu'un travail un peu important, chez nous, nous oblige à faire appel à une main d'œuvre extérieure, l'aide d'un voisin bienveillant par exemple. Cela ne pourrait présenter d'intérêt que pour faciliter la vie à tout le monde, Il ne viendrait jamais à l'idée de personne de faire appel à une aide qui puisse créer plus de difficultés qu'on n'en a déjà à surmonter. Ou si c'était le cas, à propos d'un voisin par trop maladroit, on éviterait de s'encombrer d'une surcharge. Tout cela semble procéder du plus simple bon sens.

Or qu'en est-il en fait par rapport à la création de l'Europe ? on pourrait espérer qu'en s'unissant les divers composants vont pouvoir se simplifier la vie en se partageant les efforts, et s'il existe des démocraties dont on puisse espérer qu'elle tendent à l'amélioration du fait social, l'occasion est rêvée d'espérer...

Ce n'est pas tout à fait ce langage que l'on entend. Pour survivre nous dit-on dans ce nouvel âge d'or, il va falloir enfin devenir vraiment compétitifs, enfin il va falloir devenir les meilleurs, et non seulement les meilleurs mais encore les plus forts. Et si c'est les autres qui deviennent plus forts qu'est ce qui va se passer ? Qu'on me laisse le droit d'être inquiet.

D'autant qu'une certaine forme de collaboration s'est déjà instaurée et qui me paraît dans la bonne ligne du processus. On s'est déjà aperçu par exemple que des entreprises allemandes venaient épandre des masses de déchets mal identifiés sur des territoires de communes dont le Conseil était bien content de ramasser quelques subsides. Conséquence bien prévisible de la pauvreté. Côté tourisme les choses sont du même ordre, les touristes à monnaie forte investissent le midi en amenant avec eux, entre autres précautions, leur nourriture, car ils n'ont pas tellement confiance à l'agro-alimentaire français. Mais par contre ils abandonnent toutes leurs poubelles en partant. C'est encore le contribuable français qui règle la facture des infrastructures

d'accueil dans les communes, ainsi que l'installation des autoroutes qui vont permettre aux allemands, aux britanniques, aux italiens et bientôt aux centr'européens d'aller dépenser leur fric en Espagne.

Il reste de nombreuses traces des chemins qui menaient jusqu'à Saint Jacques de Compostelle. Des troupes de pèlerins avaient alors le choix entre se faire couper la bourse et le cou par des bandits de grands chemins qui les attendaient en salivant, à chaque tournant de montagne, ou encore se faire accompagner par des troupes armées réparties sur tout le parcours et qui, bien entendu, se faisaient rétribuer leur protection et qui leur coupaient aussi un peu le cou s'ils étaient réticents. Les méthodes ont évolué mais on ne peut pas dire qu'elles se soient détériorées. On ne coupe plus les cous, simplement parce que le social ayant évolué, au lieu d'un seul pèlerinage dans sa vie, on peut désormais en faire un par an.

Tout ceci ne restreint pas pour autant mon inquiétude, d'autant que je vois parfaitement l'intérêt des multinationales dans l'affaire. Mais celui du simple citoyen que je suis ?

Poursuivons le raisonnement. Une Europe économique, car c'est bien de cela qu'on nous rebat les oreilles, va s'organiser, nous affirme-t-on la main sur le cœur, selon des principes économiques. Ce n'est pas ici que Lapalisse risque le viol. D'autant moins qu'il s'agit d'une notion largement ambiguë, si ce n'est équivoque. À qui peut profiter le crime ?

Peut-être faudrait-il préciser. Or, comme par hasard, ce qui s'impose d'emblée à nos cervelles technocratiques c'est la réduction des coûts. Opération bien difficile dans nos civilisations aux lois sociales durement arrachées et fortement structurées. Tandis que la présentation d'un argument d'autant plus péremptoire qu'il est issu d'un intérêt visiblement "supérieur" doit enfin pouvoir jouer le rôle de ce qu'il faut de vaseline pour que tout ce qu'il faut s'enclanche là où il faut.

Inutile de s'attendre à ce qu'on nous présente l'Europe comme une mise en commun d'efforts qui va simplifier la vie à tout le monde, en permettant par exemple, on peut rêver tout de même, que la suppression des gaspillages tels que les excès publicitaires, les campagnes électorales, l'armée, la douane puisque nous sommes tous amis, ainsi de suite, va sérieusement alléger la charge. Eh bien non! on commence par créer une armée commune, encore plus coûteuse. On commence par des accords de police encore plus répressifs afin que les "citoyens" mécontents ne puissent plus s'évader dans un pays voisin. On va étaler les coûts au niveau le plus bas. Or, ce qui coûte le plus cher dans nos civilisations sociales c'est encore le travailleur... considéré comme une unité de production, bien entendu, l'être humain lui, possède une valeur bien moindre. Je n'aurai pas le mauvais goût d'exhiber à l'appui d'un tel persiflage l'exemple du sang qu'on savait infecté du sida et

commercialisé tout de même, car, horreur suprême, l'entreprise risquait la faillite.

La pire notion qu'on ait jamais pu inventer, la compétition, s'est révélée un excellent moyen pour fabriquer l'esclave de type parfait. L'individu ainsi engagé dans l'effort qui l'aveugle perd totalement son esprit critique. Le meilleur ouvrier de France, un métallo, est fier de travailler sur un sous-marin atomique, alors que son grand-père, il l'a oublié, tentait de s'opposer à la coulée des tourelles de chars. Et plus il en a bavé, moins il supporte que d'autres refusent de s'engager dans une résistance à la crétinisation. Il en oublie même que le suppôt du pouvoir, quelque soit la couleur politique dont il se pare n'est pas là pour l'écouter ou pour l'aider, comme il le prétend, mais pour exercer le pouvoir: pour le réduire à l'esclavage. Quand en 81, la "gauche" a succédé à la "droite", la minute de détente a été suffisante pour que le citoyen se fasse placer quelques banderilles dont il se demande encore d'où elle viennent, je parle des citoyens idéalistes, bien sur, pas de ceux qui avaient encore mémoire de l'origine de la sale guerre d'Algérie, de Ramadier et sa vignette, maintenant la poll tax, etc. Et ce n'est pas spécialement parce qu'elle se déclare de "gauche" qu'Edith est particulièrement dangereuse, mais simplement parce qu'elle a atteint au "pouvoir" et qu'elle l'exerce. Tout simplement.

Mais sommes nous obligés de nous laisser faire ?

C'est pourquoi, si Lapalisse ne craint plus rien, le simple citoyen me paraît au contraire avoir tout à craindre. Et si je peux me permettre un conseil, il me semble important de regarder le pouvoir actuel bien en face, et de faire très attention en se retournant.

E. Bianco



# SYSTEME DE CONDUITE DE PROCESSUS

Sami Hilala

Subject Classification Informatics: C53 C54 D31 D41 .

**RESUME.** Partant d'un langage de conduite de processus, l'auteur élabore une structure de système capable, couplé avec la machine universelle de ce langage, de piloter n'importe quel programme. Il se base sur la notion d'autojectivité pour organiser les rapports entre les divers organes qui se trouvent ainsi mis en relation.



## SYSTEME DE CONDUITE DE PROCESSUS (SCP)

### **Présentation.**

La compétitivité à toujours été l'une des conditions d'existence des entreprises. La capacité de fabriquer des produits de qualité optimale pour un coût minimal en est une composante essentielle;

- diminuer les coûts;
- améliorer le rendement de production;
- maîtriser la qualité;

sont les objectifs de tout chef d'entreprise à travers différentes actions, parmi lesquelles l'automatisation et la conduites de processus occupent une place primordiale.

L'extension croissante des applications de l'électronique, la diminution fantastique du prix des composants, la naissance et le développement des micro-processeurs, ont permis l'utilisation massive des automates programmables, qui sont conçus dans une optique particulière pour des automaticiens.

L'un des inconvénients que présentent de tels ensemble, c'est l'impossibilité d'échange avec des ordinateurs, on se servira d'un ordinateur pour concevoir ou programmer les automates programmables.

Il nous paraît intéressant d'introduire l'ordinateur tel que nous le connaissons mais en le munissant alors d'un système adapté;

On peut appeler ce nouvel instrument "**Automates Programmables**".

Nous présentons dans cet article, la description d'un système, le SCP, et d'un dérouleur, destinés à assurer le contrôle de tout programme écrit en LCP<sup>1</sup> compilé.

---

<sup>1</sup> Langage de Conduite de Processus



# SYSTEME DE CONDUITE DE PROCESSUS (SCP)

## INTRODUCTION:

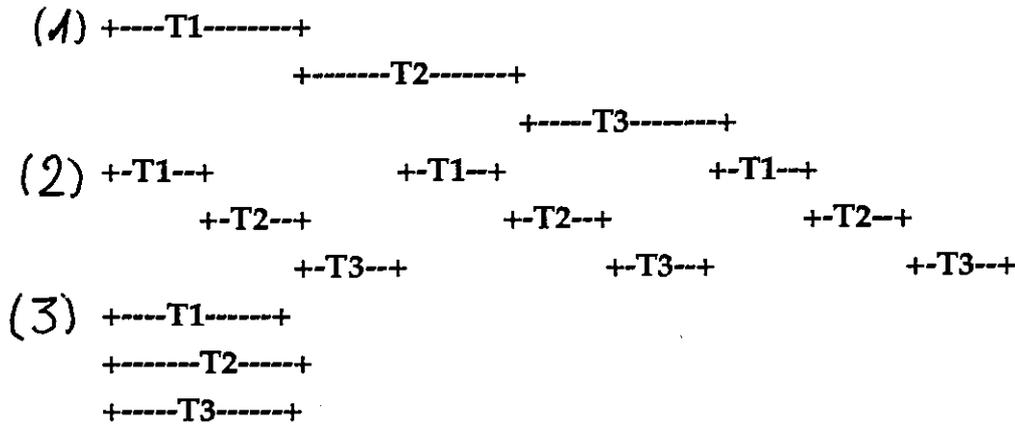
Dans le déroulement d'un ensemble d'opérations, afin de maîtriser la répartition des charges, nous allons séparer les opérations de traitement de la commutation globale, c'est-à-dire des opérations de distribution des calculs.

Une des propriétés essentielles du LCP réside dans la possibilité de décrire le déroulement parallèle de plusieurs actions distinctes. Et, au déroulement, peuvent apparaître, dynamiquement, un nombre quelconque et évolutif de telles tâches. C'est donc au déroulement que seront pris en compte, traitements de variables et commutations locales. C'est une machine universelle qui est chargée, à la lecture des informations correspondantes, codées, d'effectuer les calculs.

On fait en sorte que cette machine ne puisse exécuter, à la fois qu'un seul acte qui soit fini-borné. Les actes sont alors enchaînés les uns aux autres, dans un déroulement, par le moyen d'un programme chef d'orchestre baptisé système de conduite de processus.

Ce SCP est organisé pour établir une rotation dans la distribution des quanta de travail, de telle sorte qu'il insère le dérouleur, chaque fois, pour un acte fini-borné, et successivement pour chacune des branches traitées en parallèle à l'instant donné. De même que la Machine Universelle, par définition est unique quelque soit le programme LCP qu'on lui confie, ce système est conçu pour être unique et définitif, quelque soit le programme LCP dont il faut diriger le déroulement, c'est ce que nous allons essayer d'expliquer.

Considérons trois tâches distinctes T1, T2 et T3. L'exécution de l'ensemble de ces trois tâches peut se dérouler de diverses manières, caractérisées par différentes traces temporelles [ KRAKOWIAK 85] :



Les différents schémas se caractérisent comme suit :

- (1) : chaque tâche est entièrement exécutée avant passage à la suivante. Cette situation correspond à une exécution séquentielle des tâches ;
- (2) : on exécute alternativement une suite d'instructions de T1, puis une suite d'instructions de T2, puis une autre de T3, et ainsi de suite jusqu'à la fin des trois tâches. Ceci correspond à une exécution en pseudo-parallèle, c'est-à-dire une exécution où une seule action peut être en cours à un instant donnée;
- (3) : l'exécution des tâches est simultanée. Ceci correspond à une situation de parallélisme réel qui nécessite la présence de plusieurs microprocesseurs.

Nous avons choisi le schéma de type (2) pour notre étude, tout simplement pour pouvoir utiliser le travail de nature séquentielle des processeurs ordinaires.

## LA DISTRIBUTION DES TACHES :

Le déroulement en parallèle<sup>2</sup> de n tâches s'effectue de la manière suivante :

En fait on déroule un petit morceau de chacune par rotation. Pour réaliser ce découpage en vue d'une distribution du travail, on structure le déroulement de façon qu'au bout d'une quantité fini borné, la tâche retourne d'elle même au système, qui n'est autre que le distributeur de quanta.

On définit un quantum de travail comme quelque chose de fini-borné dont on connaît a priori l'importance et qui correspond par exemple pour une tâche en cours d'exécution, au déroulement d'une instruction, ou au déroulement d'une instruction à durée illimitée. Il y a ainsi des instructions dont on sait par avance qu'elles ait une durée bornée et suffisamment petite et d'autres que le système lance et dont il sait périodiquement constater l'état d'avancement- ce qui ne risque pas de bloquer de la sorte le reste du travail .

Les travaux du système et des Dérouleurs sont parfaitement séparés, les Dérouleurs sont préstructurés, ce qui signifie que le travail du système se réduit à la seule gestion des tâches, aussi bien de traitement que d'échanges.

Du point de vue de l'organisation, le système fonctionne comme un programme dans lequel s'insèrent des sortes de procédures dont le temps de déroulement est strictement borné. Le travail complet de la tâche étant réalisé par l'insertion d'autant de ces procédures qu'il est nécessaire. Le système ne peut plus être bloqué par un afflux quelconque d'information puisqu'il demeure toujours maître de la situation.

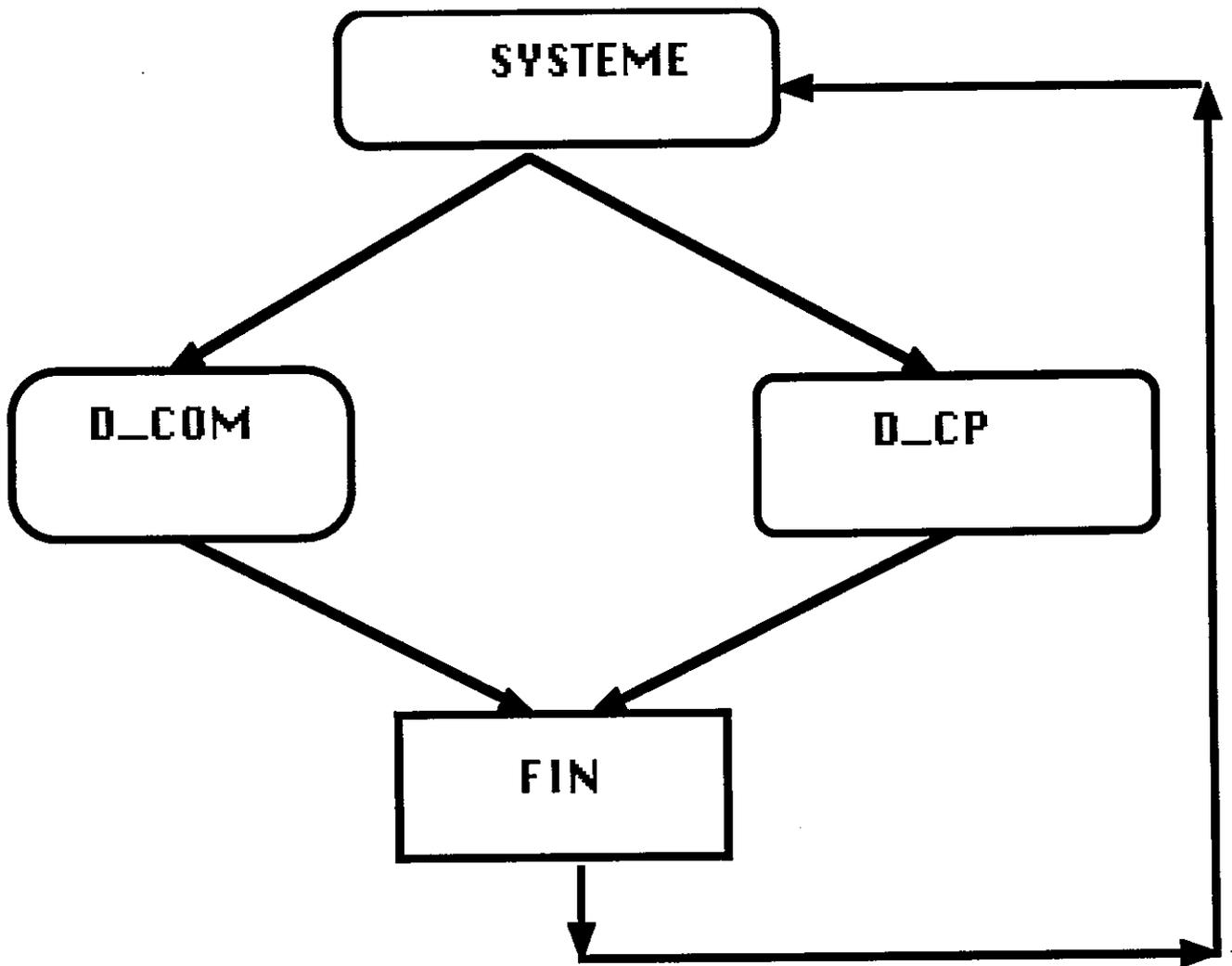
On distingue deux sortes de dérouleurs utilisés dans notre étude:

- **le dérouleur des communications (D\_COM)**, qui assure le traitements du type conversationnels; ces derniers qui impliquent un échange d'information entre l'unité central et d'autres dispositifs d'échange (micro-ordinateur, clavier, horloge temps réels, etc.. ).

- **le dérouleur de conduite de processus (D\_CP)**, qui assure le déroulement des tâches par interprétation du code du programme écrit en LCP.

---

<sup>2</sup> Pour la suite, parallèle correspond à pseudo-parallèle



Le rôle du système consiste à demander au dérouleur de prendre en compte le code de l'instruction en cours, de préparer le calcul de la suivante, et de retourner au système, qui répète la même opération pour la tâche suivante. Il est donc absolument nécessaire que les limites de l'encombrement, temps et espace, de l'instruction soient rigoureusement déterminées .

Mais le rôle du système ne s'arrête pas là, il examine l'état du retour du dérouleur, afin de prendre en compte les instructions d'insertion ou d'élimination des tâches déroulées par celle-ci.

**Etat de retour du système :**

Le système examine l'état de retour des dérouleurs, afin de prendre en compte lui-même certains traitements qui se rattachent à certaines instructions, par exemple, l'insertion des tâches, l'enlèvement d'une tâche dans la file de déroulement ou la réorganisation de la mémoire :

**= Insertion d'une ou plusieurs tâches :**

le dérouleur D\_CP, signale au système la prise en compte de l'instruction d'insertion des sections (tâches) en parallèle (OU/ET), ou l'insertion simple. Les adresses des sections sont mises par le D\_CP, dans la file de retour "f\_ret ", à la suite du nombre des sections. Le rôle du système consiste, dans ce cas, à faire apparaître les images des sections dans la file de déroulement (f\_der), à les activer ces sections, puis, à mettre l'état de la section courante à "attente";

**= Enlèvement d'une tâche :**

Chaque fois que le système arrête une tâche, il l'enlève de la file de déroulement c'est-à-dire qu'il met son état à "terminé", ensuite, il réorganise la file de déroulement;

**= Réorganisation de la mémoire (file de déroulement) :**

Chaque fois que le système enlève une tâche de la file de déroulement , il réorganise cette file pour remplir le trou. Cette opération peut se faire de plusieurs manières :

Soit en décalant toutes les tâches qui se trouvent à droite de la tâche enlevée;

Soit en plaçant la dernière tâche, à la place de la tâche enlevée.

C'est la deuxième solution que nous choisissons. Mais la réorganisation de la mémoire ne s'arrête pas là, on change dans toutes les tâches appelées par celle qu'on vient de déplacer, le N° d'appel par le nouveau N°; car le N° de la tâche placée à la place de la tâche enlevée, prend le N° de la tâche courante (tâche enlevée).

Configuration du système :

Le code du système ainsi que celui du dérouleur sont implantés à demeure (p.ex. sur PROM ou EPROM), le reste de mémoire sert à implanter le code du programme à dérouler avec sa configuration et une partie de mémoires sert au système pour gérer les images de tâches:

Code syst	Vars. syst	f_dér	f_ret	code prog.	Vars. prog.
I_____	I_____	I_____	I_____	I_____	I_____

D'abord on a la configuration du système, suivie par la file de déroulement (f\_der), puis la file de retour (f\_ret), suivie par le code du programme en C.P<sup>1</sup>, à dérouler, enfin la table de données (les variables) :

= configuration du système :

d	f_ret	f_der	NB_T	N°Tc	der_c	dp	P	Pc	T_v	dVs	dVsi	dVD
I_I	I_____	I_____	I_____	I_____	I_____	I_____	I_____	I_____	I_____	I_____	I_____	I_____
I	I <sub>0</sub>	1	2	3	4	5	6	7	8	9	10	11 12

et_au	Cd	Ct	Cpt	Adr1	Adr2	Adr3	24	Aux1	Aux2	ADR	V0	...	Vn
I_I	I_____	...	I_____										
14	15	16	17	18	19	20	21	22	23	24			
										1	.....		

d : adresse de la première case libre;

f\_ret : début de la file qui contient l'état de retour du dérouleur;

f\_der : début de la file de déroulement ;

NB\_T : nombre de tâches à dérouler en parallèle à un instant donné;

N°Tc : Numéro de la tâche en cours de déroulement;

der\_c : adresse de la tâche courante;

---

<sup>1</sup> C.P. : Conduite de Processus.

**dp** : adresse de début du code du programme écrit en C.P. à dérouler, par rapport à l'origine ; zéro pour "dp" indique que le code de programme est absent;

**P** : adresse du début de l'instruction à dérouler;

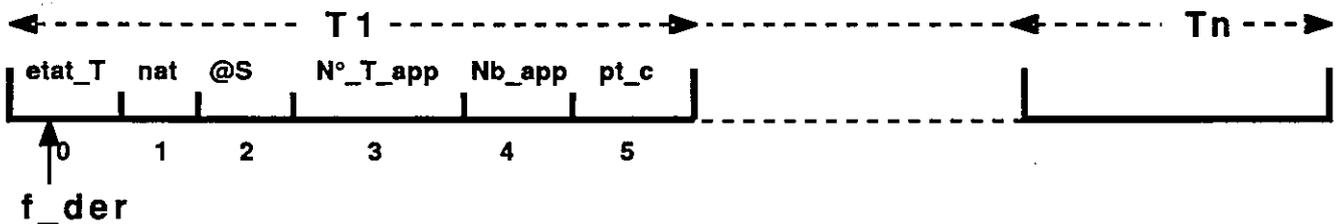
**Pc** : pointeur de l'adresse de l'instruction en cours, pour l'exploration du code d'une même instruction;

**d** : adresse du début de la table des données (les variables);

**dVs, dVsi, dVD** : adresse du début des variables (Sortie, Sortie\_I et Durée respectivement), dans la table des variables ;

**eta\_aut, Cd, Ct, Cpt, ADR** : paramètres concernant le dérouleur des communications (D\_COM) (Cf. IV.4.3.2).

**f\_der** = file de déroulement (f\_der) : destinée aux dérouleurs, et qui contient pour chaque tâche à dérouler 6 paramètres ;



**etat\_T** : état du déroulement de la tâche "Ti". Les valeurs que peut prendre `etat_T` sont :

- "active" : mise en place par le système au moment de l'appel de la tâche;

- "attente" : le système remplace la valeur "active" par "attente", quand la tâche courante appelle une ou plusieurs autres tâches;
- "terminé" : quand la D rencontre le fin de la section (tâche) en déroulement, le système remplace la valeur "active" par "terminé";

**nat** : nature de la tâche "Ti". L'ensemble des valeurs prises par "nat" en fonction de la nature de la tâche :

Tâche	Valeur
Communication	"com"
Dérouler_et	"der_et"
Dérouler_ou	"der_ou"
Insérer_simple	"inser"

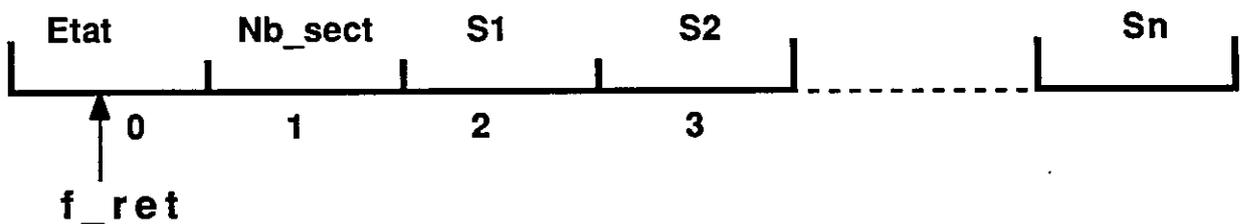
**@S** : adresse de la section par rapport à "dp";

**N°\_T\_app** : numéro de la tâche appelante, dans la file de déroulement;

**NB\_app** : nombre de tâches appelées par cette tâche;

**Pt\_c** : pointeur d'instruction courante ;

= **file de retour (f\_ret)** : destinée à l'état de retour des opérations en cours du dérouleur :



**Etat** : état du retour du dérouleur en cours, les valeurs possibles sont :

"rien" : mise en place par le système avant l'appel d'un dérouleur.

"der\_ou" / "der\_et" : le D\_CP, signal au système la rencontre de l'instruction "dérouler\_ou" (ou "dérouler\_et"), pour insérer les sections appelées par cette instruction;

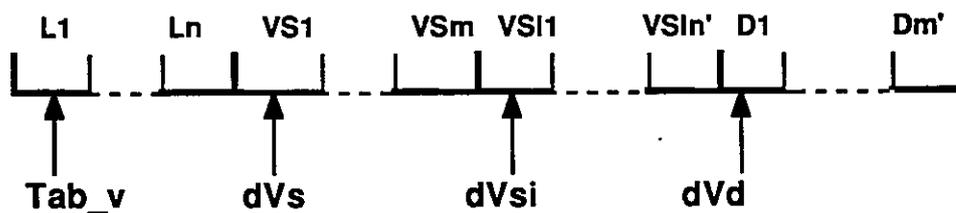
"inser" : même traitement que le cas ci-dessus, mais pour une seule insertion;

"fin\_sect" : le D\_CP, signal au système la fin de la section en cours;

"fin\_prog" : le D\_CP, signal au système la fin du programme écrit en L.C.P;

"inser\_prog" : le D\_COM, signal au système le lancement d'un programme CP.

= la table des variables (Tab\_v), utilisée pour le déroulement d'un programme en L.C.P:



A l'initialisation le système doit créer les configurations. Ensuite, il initialise (mise à zéro) les divers paramètres du système, puis, active la première tâche à dérouler (la tâche de communication ).

Le système commence par la préparation de la tâche à dérouler, ensuite, il initialise l'état de retour, puis, passe à l'un des deux dérouleurs (D\_CP, D\_COM), pour l'exécution de l'instruction de la tâche courante. Le système reprend enfin la main, pour examiner l'état de retour du dérouleur.

### Programme du système :

Procédure SYSTEME:

PARA : 3;

INDEX : 0;

LOCAL : 21 : 1, 1, 1, ..., 1;

Début

" Initialisation "

$l_3 := 1;$  NB\_T ← 1 ;

$l_4 := 0;$  N°Tc ← 0 ;

$l_5 := l_2;$  der\_c ← f\_der ;

$l_6 := 0;$  dp ← 0 ;

$l_9 := ;$  Tab\_v ← ;

$l_{5,0} := \text{"active"};$

$l_{5,1} := \text{"com"};$

$l_{5,2} := 0;$

$l_{5,3} := 0;$

$l_{5,4} := 0;$

$l_{5,5} := 0;$

" Fin Initialisation "

Debsys :  $l_{1,0} := \text{"rien"};$  état de retour à "rien";

**SI**  $l_4 \geq l_3$  **VERS** Init\_T;

**SI**  $l_{5,1} = \text{"com"}$  **VERS** D\_com;

**SI**  $l_{5,0} \neq \text{"active"}$  **VERS** Fin\_syst;

" préparer D\_CP"

	$l_7 := l_6 + l_{5,2};$	$p := @Si + @dp;$
	$l_7 := l_7 + l_{5,5};$	$p := p + pt_c;$
	<u>VERS</u>	D_CP;
<u>esyst1 :</u>	<u>VERS</u>	Ret_syst;
<u>Fin syst :</u>	$l_4 := l_4 + 1;$	$N^{\circ}Tc := N^{\circ}Tc + 1;$
	$l_5 := l_5 + 6;$	@Tâche := @Tâche_suivante;
	<u>VERS</u>	Debsys ;
<u>Init T :</u>	$l_4 := 0;$	$N^{\circ}Tc := 0;$
	$l_5 := l_2;$	@Tâche := @1ER_Tâche;
	<u>VERS</u>	Debsyst ;

Fin.

### Traitement de l'état de retour :

C'est le rôle du système d'examiner l'état de retour des dérouleurs, par l'intermédiaire de la variable "Etat", dans la file "f\_ret". Le système prend un certain nombre de décisions en fonction de l'état de cette variable :

- "rien" : dans ce cas, le système ne fait rien;
- "der\_ou"/"der\_et"/"inser" :

le dérouleur D\_CP, signale au système la prise en compte de l'instruction d'insertion des sections (tâches) en parallèle (OU/ET), ou l'insertion simple. Les adresses des sections sont mises par le D\_CP, dans la file de retour "f\_ret ", à la suite du nombre des sections. Le rôle du système consiste, dans ce cas, à insérer les sections dans la file de déroulement (f\_der), à les activer, puis, à mettre l'état de la section courante à "attente";

- "fin\_sect" : quand le D\_CP, rencontre la fin de la section en cours, il met l'état de retour à "fin\_sect". Le système distingue plusieurs cas en fonction de la nature de la tâche (section) en cours:

= si la tâche à été appelée par une autre, à l'aide d'une insertion simple, alors le système enlève l'image de la tâche, puis il active la tâche appelante. Le système réorganise enfin, la gestion de la mémoire (file de déroulement) (cf. page 10);

= si la tâche (Ti) à été appelée par une autre (Tj), à l'aide d'une instruction "dérouler\_ou", au moment où elle s'arrête, le système doit arrêter toutes les tâches appelés par Tj dans le "dérouler\_ou", puis il les enlève de la file de déroulement, qu'ensuite, il réorganise ;

= dans le cas ou la tâche (Ti) à été appelée par (Tj), à l'aide d'une instruction "dérouler\_et", alors le système met l'état de la tâche Ti à "terminé" quand celle-ci s'achève, puis, décrémente le nombre de tâches appelées par Tj "Nb\_app". A partir du moment où ce nombre est égal à zéro, ce qui signifie que toutes les tâches appelées par Tj ont achevé leurs exécutions, alors le système traite toutes les images des tâches achevées, comme dans le cas précédent.

- " inser\_prog " : lancement du programme CP.

- "fin\_prog" : fin du programme C.P. en cours. Le système met la valeur de communication "prog" à "libre", afin d'autoriser le déroulement d'un autre programme .

Algorithme de l'état de retour :

Ret syst :

**SI**  $l_{1,0} = \text{"rien"}$       **VERS** Fin\_ret;

**SI**  $l_{1,0} = \text{"fin_sect"}$     **VERS** er\_fs;

**SI**  $l_{1,0} = \text{"fin_prog"}$     **VERS** er\_fp;

```

SI    [1,0 = "inser_prog" VERS eins_p;

        "inser", "dér_OU", "dér_ET"

[17 := [2 + 6*[3;      Adr1 := @f_der + 6*NB_T;

[18 := [1 + 2;          Adr2 := @f_ret + 2 ;

[5,4 := [1,1 ;        Nb_app := Nb_sect;

[5,0 := "attente";    etat_Tc := "attente";

eRsuite : SI    [1,1 = 0      VERS Fin_ret;

[17,0 := " active";    etat_T := "active";

[17,1 := [1,0;         nat_T := Etat;

[17,2 := [18,0;        @S := @Si ;

[17,3 := [4;           N°_T_app := N°Tc

[17,4 := 0;            Nb_app := 0;

[17,5 := 0;            Pt_c :=0;

[3 := [3 + 1;          NB_T := NB_T + 1;

[17 := [17 + 6;

[18 := [18 + 1;

[1,1 := [1,1 - 1;      Nb_sect := Nb_sect - 1;

VERS eRsuite;

er fp : [6 := 0;      Prog := "libre" ( DP <---- 0 );

[3 := [3 - 1;

[4 := [4 - 1;

```

```

[5,0 := "terminé" ;
VERS Fin_ret;

er fs : SI [5,1 = "der_ou" VERS efs_ou;
          SI [5,1 = "der_et" VERS efs_et;
          SI [5,1 = "inser" VERS efs_ins;

VERS efret_fs;

efs ins : [17 := [2 + 6*[5,3;      Adr1 := f_der + 6*N_T_app;
           [17,0 := "active";
           [17 := [2 + 6*[4;      Adr1 := tâche à enlever ;
           [20,0 := [4;      V0 <----- N°T_C;

VERS Enlever_Tc;

eins1 : [4 := [4 - 1;

VERS efret_fs;

eins p : [17,0 := "active";      etat_T := "active";
          [17,1 := "inser" ;      nat_T := "inser";
          [17,2 := [6;      @S := @DP ;
          [17,3 := 0;      N°_T_app := 0
          [17,4 := 0;      Nb_app := 0;
          [17,5 := 0;      Pt_c := 0;
          [3 := [3 + 1;      NB_T := NB_T + 1;

VERS Fin_ret ;

efs ou : [17 := [2 + 6*[5,3;      Adr1 := f_der + 6*N°_T_app;

```



```

[20,0 := N°_tâche à enlever;
[18 := [2 + 6*([3-1];      Adr2 := f_der + 6*Nb_T;
[17,0 := [18,0;
[17,1 := [18,1;
[17,2 := [18,2;
[17,3 := [18,3;
[17,4 := [18,4;
[17,5 := [18,5;
[3 := [3 - 1;      Nb_T := Nb_T - 1;

      " réorganisation de la f_der"

[18 := [2;
[20,1 := 0;
esr1 :   SI   [20,1 = [3      VERS freorg;
          SI   [18,3 = ([3 - 1);  VERS ech;
esr2 :   [18 := [18 + 6;
          [20,1 := [20,1 + 1;
          VERS esr1;
ech :   [18,3 := [20,0;
          VERS esr2;
freorg : SI   [5,1 = "inser"    VERS eins1;
          VERS eenoul;

```

Fin Enlever Tc

### LE DEROULEUR DE CONDUITE DE PROCESSUS (D\_CP):

On va décrire notre dérouleur (D\_CP), avec la Procédure Formelle [BIANCO 82-83 ]. Ceci implique qu'on construit une procédure qui va s'appliquer sur une configuration de structure standard. On suppose que les codes des diverses instructions qui composent le programme écrit en L.C.P., sont alignés en mémoire les uns à la suite des autres, à partir d'une case de la configuration système, repérée par une distance "dp" de l'origine "I".

Le système initialise l'adresse "p", qui nous permet d'atteindre le code de l'instruction à exécuter de la tâche (section) en cours d'exécution. On aura besoin d'une adresse "pc" initialisée avec "p", mais utile pour l'exploration du code d'une même instruction.

#### Aiguillage de la "D\_CP":

On a vu dans le chapitre III, que la forme du code des instructions du langage image est comme suit :

( Code\_op, N1, N2, N3) (A1) (A2) (A3) . . . (An) .

L'aiguillage du dérouleur D\_CP, commence par analyser le code de l'opération "code\_op". "pc", désigne la première case du code de l'instruction qu'analyse l'aiguillage.

**ENTREE:**

<b>SI</b> [7,0 = "Prog"	<b>VERS</b> edéb_prog ;
<b>SI</b> [7,0 = "Fin"	<b>VERS</b> efin_prog ;
<b>SI</b> [7,0 = "Prog"	<b>VERS</b> edéb_prog ;
<b>SI</b> [7,0 = "=="	<b>VERS</b> edpeg ;
<b>SI</b> [7,0 = ":-"	<b>VERS</b> edpm ;
<b>SI</b> [7,0 = ":=+"	<b>VERS</b> edpp ;
<b>SI</b> [7,0 = "=="	<b>VERS</b> edpmul ;
<b>SI</b> [7,0 = ":=not"	<b>VERS</b> ednot ;
<b>SI</b> [7,0 = "cond="	<b>VERS</b> econdeg ;
<b>SI</b> [7,0 = "cond=+"	<b>VERS</b> econdegp ;
<b>SI</b> [7,0 = "cond=-"	<b>VERS</b> econdegm ;
<b>SI</b> [7,0 = "cond=."	<b>VERS</b> econdegmul ;
<b>SI</b> [7,0 = "cond=not"	<b>VERS</b> econdegnot ;
<b>SI</b> [7,0 = "cond<>"	<b>VERS</b> econdd ;
<b>SI</b> [7,0 = "cond<>+"	<b>VERS</b> econddp ;
<b>SI</b> [7,0 = "cond<>-"	<b>VERS</b> econddm ;
<b>SI</b> [7,0 = "cond<>."	<b>VERS</b> econddmul ;
<b>SI</b> [7,0 = "cond<>not"	<b>VERS</b> econddnot ;
<b>SI</b> [7,0 = "Dérouler_ET"	<b>VERS</b> edéret ;
<b>SI</b> [7,0 = "Dérouler_OU"	<b>VERS</b> edérou ;
<b>SI</b> [7,0 = "Sect"	<b>VERS</b> eins ;
<b>SI</b> [7,0 = "Sect_cond"	<b>VERS</b> einscond ;
<b>SI</b> [7,0 = "Répéter"	<b>VERS</b> erépéter ;
<b>SI</b> [7,0 = "Répéter_SI"	<b>VERS</b> erépétersi ;
<b>SI</b> [7,0 = "Fin_Répét"	<b>VERS</b> efinrédépét ;
<b>VERS</b> erreur_code;	

A chacune des étiquettes auxquelles renvoie T1, se traite complètement le code d'une instruction. Après avoir achevé ce calcul, "pc" aura évolué de façon à contenir le nombre de cases du code de l'instruction exécutée de la tâche en cours, et on ajoute la valeur de "pc" à "pt\_c", pour désigner la première case de l'instruction suivante, par rapport à l'adresse de la tâche, et on se renvoie à la fin de l'aiguillage (Fin\_D\_CP).

### LE DEROULEUR DE COMMUNICATION (D COM):

Notre dérouleur de communication est sous contrôle du système, il communique avec le monde extérieur, à l'aide de dispositifs externes tels que, clavier, afficheurs, système intelligent.

Un langage de communication est nécessaire pour pouvoir communiquer avec de tels dispositif.

### Le langage de communication :

Notre langage de communication est très simple, il est défini comme suit :

pour évoquer le mot pointage (d'adresse). Cette opération étant effectuée, le dispositif de communication est en attente de la lecture des chiffres hexadécimaux qui formeront l'adresse pointée;

pour mémoriser. Lorsque l'adresse est entièrement constituée, et si l'opérateur envoie le mot "M", alors le système fait apparaître le contenu de la case correspondant à l'adresse. Ici deux cas se présentent:

- soit l'opérateur ne fait que lire le contenu mémoire, pour le vérifier par exemple et ne souhaite pas le modifier. Lorsqu'il envoie de nouveau le mot "M", il y aura incrémentation automatique de l'adresse et le contenu de cette nouvelle adresse apparaîtra. En pratiquant ainsi plusieurs fois de suite, l'opérateur pourra lire le contenu d'une zone mémoire;
- soit l'opérateur veut modifier ou tout simplement charger à une valeur donnée, la case mémoire pointée. Dans ces conditions, il doit envoyer

les chiffres hexadécimaux correspondants. Lorsque la donnée est ainsi introduite, l'opérateur doit envoyer le mot "M" pour, d'une part, introduire le mot de données en mémoire et, d'autre part, incrémenter l'adresse;

pour lancer l'exécution. Une fois le code du programme en conduite de processus introduit en mémoire, l'opérateur peut le faire dérouler. pour cela il sélectionne d'abord le pointage d'adresse "P" et il introduit l'adresse en hexadécimal de la première instruction exécutable. Lorsque l'adresse est complète, il suffit d'envoyer le mot "L" pour que le système lance le programme<sup>1</sup>;

pour "Initialisation".

#### Entrée et exécution d'un programme :

L'introduction du code d'un programme en hexadécimal dans la mémoire (VIVE) de l'unité centrale se passe de la manière suivante:

il faut envoyer :

la commande "I", pour l'initialisation;

la commande "P", positionnement d'adresse;

quatre chiffres hexadécimaux représentatifs de l'adresse de la première case où doit être implanté le programme;

la commande "M", pour mémorisation; sur les afficheurs de données, apparaissent deux chiffres hexadécimaux représentatifs de l'octet de donnée présent à l'adresse préalablement pointée.

les deux chiffres hexadécimaux représentatifs de l'octet de données à entrer. pour en réaliser la mémorisation, il est nécessaire d'effectuer l'opération suivante;

la commande "M", l'adresse est incrémentée de 1 et une nouvelle valeur apparaît sur les afficheurs de données. c'est le contenu de l'octet suivant en mémoire;

---

<sup>1</sup> Tant que le programme de conduite de processus n'est pas achevé, on ne peut pas lancer l'exécution d'un autre.

les deux chiffres hexadécimaux représentatifs de l'octet suivant du programme;

de nouveau la commande "M" et revenir à l'étape 5, jusqu'au dernier octet à introduire.

Pour lancer l'exécution du programme, on envoie d'abord la commande I, pour l'initialisation, puis la commande "P", positionnement d'adresse, puis, on envoie les quatre chiffres hexadécimaux représentatifs de l'adresse de la première case où doit être implanté le programme, enfin on lance l'exécution par la commande "L".

**Exemple :**

Soit le code d'un programme donné en hexadécimal comme :

AF 03 D1 5F 32 , on va l'introduire à partir de l'adresse (3AF6)H :

Commande	ADRESSES	DONNEES
====> <sup>3</sup> I	0	0
====>p	0	0
====>3	0	0
====>A	A	0
====>F	A	F 0
====>6	A	F 6
====>M	A	F 6
====>A	A	F 6
		X <sup>4</sup> X
		A X

<sup>3</sup> signifie introduire la commande.

<sup>4</sup> l'écriture X represent une valeur inconnue.

====> F	A	F 6	A F
====> M	A	F 7	X X
====> 0	A	F 7	X
====> 3	A	F 7	3
====> M	A	F 8	X X
====> D	A	F 8	D X
====> 1	A	F 8	D 1
====> M	A	F 9	X X
====> 5	A	F 9	X
====> F	A	F 9	F
====> M	A	F A	X X
====> 3	A	F A	X
====> 2	A	F A	2
====> M	A	F B	X X

La procédure de la relecture de notre programme est la suivante :

Commande	ADRESSES		DONNEES
====> I	0	0	0
====> p	0	0	0
====> 3	0	0	0
====> A	A	0	0

====> F	A	F 0	0
====> 6	A	F 6	0
====> M	A	F 6	A F
====> M	A	F 7	3
====> M	A	F 8	D 1
====> M	A	F 9	F
====> M	A	F A	2

Si au cours de la relecture, un octet s'avère erroné, il suffit d'envoyer les deux chiffres hexadécimaux corrects, puis envoyer la commande de mémorisation "M" de nouveau. Ainsi l'ancien octet est écrasé par le nouveau.

Pour lancer l'exécution de ce programme :

Commande	ADRESSES	DONNEES	
====> I	0	0	0
====> p	0	0	0
====> 3	0	0	0
====> A	A	0	0
====> F	A	F 0	0
====> 6	A	F 6	0
====> L	A	F 6	0

**Le dérouleur de communication :**

Notre langage étant défini, on peut maintenant réaliser le dérouleur du code de ce langage. Pour cela on peut utiliser un automate d'états fini. Ici on utilisera la forme d'un graphe dont les sommets sont appelés états, où les flèches, appelées "transition", sont étiquetées par un symbole (code) du vocabulaire d'entrée<sup>2</sup>.

Pour notre langage on peut avoir :

---

<sup>2</sup> Vocabulaire d'entrée = ( "P", "M", "L", "I", "HEXA"), où "HEXA" = ("0"... "9", "A"... "F").



Un moyen simple de programmer un tel automate est de construire une table des transitions et d'écrire un dérouleur de table :

**table des transitions :**

Vocab. Etat	"P"	"I"	"M"	"L"	"HEXA"	aut
Etat0	Etat1	Etat0	Etat4	EtatER	EtatER	EtatER
Etat1	Etat1	Etat0	Etat4	EtatER	Etat2	EtatER
Etat2	Etat1	Etat0	Etat4	Etat3	Etat2	EtatER
Etat3	Etat1	Etat0	Etat4	EtatER	EtatER	EtatER
Etat4	Etat1	Etat0	Etat4	EtatER	Etat5	EtatER
Etat5	Etat1	Etat0	Etat4	EtatER	Etat5	EtatER
EtatER	Etat1	Etat0	EtatER	EtatER	EtatER	EtatER

On remarque :

= qu'on passe de n'importe quelle état à l'état "Etat1", lorsqu'on lit la commande "P", au cours de ces transitions on doit initialiser les adresse et les diverses paramètres;

= on passe de l'état "Etat2" vers l'état "Etat4", lorsqu'on lit la commande "M", pour lire le contenu de la mémoire à l'adresse pointée par "P";

= etc ...

Ceci nous amène à construire une table des actions associées aux transitions:

**table des actions :**

Vocab.	"P"	"I"	"M"	"L"	"HEXA"	aut
Etat0	a2	a1	a3	a5	a5	a5
Etat1	a2	a1	a3	a5	a4	a5
Etat2	a2	a1	a3	a6	a4	a5
Etat3	a2	a1	a3	a5	a5	a5
Etat4	a2	a1	a7	a5	a8	a5
Etat5	a2	a1	a9	a5	a8	a5
EtatER	a2	a1	a5	a5	a5	a5

**Algorithme du dérouleur D\_COM :**

Les paramètres utilisés pour le dérouleur (D\_COM) sont (Cf. IV.2.2):

- "etat\_aut" : pour l'état précédente de l'automate;
- "Cd" : contient la commande envoyer;
- " Ct " : désigne la valeur de l'adresse courante;
- " ADR " : pointeur de l'adresse courante;
- "Cpt " : paramètre de contrôle.

D COM :**" Dérouleur de communication "** $l_{14} := \text{LIRE\_Com} ;$ **SI**  $l_{14} = \text{"I"}$  **VERS** ea1 ;**SI**  $l_{14} = \text{"P"}$  **VERS** ea2 ;**SI**  $l_{13} = \text{"Etat0"}$  **VERS** eEtat0 ;**SI**  $l_{13} = \text{"Etat1"}$  **VERS** eEtat1 ;**SI**  $l_{13} = \text{"Etat2"}$  **VERS** eEtat2 ;**SI**  $l_{13} = \text{"Etat3"}$  **VERS** eEtat3 ;**SI**  $l_{13} = \text{"Etat4"}$  **VERS** eEtat4 ;**SI**  $l_{13} = \text{"Etat5"}$  **VERS** eEtat5 ;**VERS** Fin\_M\_Com ;**eEtat0 :** **SI**  $l_{14} = \text{"M"}$  **VERS** ea3 ;**VERS** ea5 ;**eEtat1 :** **SI**  $l_{14} = \text{"M"}$  **VERS** ea3 ;**SI**  $l_{14} = \text{"HEXA"}$  **VERS** ea4 ;**VERS** ea5 ;**eEtat2 :** **SI**  $l_{14} = \text{"M"}$  **VERS** ea3 ;**SI**  $l_{14} = \text{"L"}$  **VERS** ea6 ;**SI**  $l_{14} = \text{"HEXA"}$  **VERS** ea4 ;**VERS** ea5 ;**eEtat3 :** **SI**  $l_{14} = \text{"M"}$  **VERS** ea3 ;

```

VERS ea5 ;
eEtat4: SI [14 = "M" VERS ea7 ;
SI [14 = "HEXA" VERS ea8 ;
VERS ea5 ;
eEtat5: SI [14 = "M" VERS ea9 ;
SI [14 = "HEXA" VERS ea8 ;
VERS ea5 ;
ea1: [23 := 0 ; ADR <--- 0 ;
[15 := 0 ; Ct <---- 0 ;
[16 := 0 ; Cpt <---- 0 ;
[13 := "Etat0" ;
VERS Fin_M_Com ;
ea2: [23 := 0 ; ADR <--- 0 ;
[15 := 0 ; Ct <---- 0 ;
[16 := 0 ; Cpt <---- 0 ;
[13 := "Etat1" ;
VERS Fin_M_Com ;
ea3: [16 := 0 ; Cpt <--- 0 ;
[15 := LIRE ([23) ;
ENVOIE ([15) ;
[13 := "Etat4" ;

```

```

VERS Fin_M_Com ;
ea4: SI [16]=0 VERS eadr1 ;
SI [16]=1 VERS eadr2 ;
SI [16]=2 VERS eadr3 ;

[23] := [23] ET (FFF0)H ;
[23] := [23] + [14] ;
[16] := 0 ;

VERS esuite ;
adr3: [23] := [23] ET (FF0F)H ;

[14] := [14] * 24 ;
[23] := [23] + [14] ;
[16] := 3 ;

VERS esuite ;
adr2: [23] := [23] ET (F0FF)H ;

[18] := [14] ;
[18] := [18] * 28 ;
[23] := [23] + [18] ;
[16] := 2 ;

VERS esuite ;
adr1: [23] := [23] ET (0FFF)H ;

[18] := [14] ;

```

```

[18 := [18 * 212;
[23 := [23 + [18;
[16 := 1;
esuite : ENVOIE ([23]);

[13 := "Etat2";

VERS Fin_M_Com ;

ea6 : SI [7 <> 0 VERS ea5 ;

[6 := [23 ; dp <--- ADR

VERS Fin_M_Com ;

ea7 : [23 := [23 + 1 ; ADR <--- ADR + 1 ;

[15 := LIRE ([23]);
ENVOIE ([15]);

[13 := "Etat4" ;

VERS Fin_M_Com ;

ea8 : SI [16 = 0 VERS emot1 ;

[16 := 0 ; Cpt <--- 0 ;

[15 := [15 ET (F0)H ;

[15 := [15 + [14 ;

VERS esa8 ;

emot1 : [15 := [15 ET (0F)H ;

[14 := [14 * 24;

```

```

[15 := [15 + [14;
[16 := 1;
esa7: ENVOIE ([15);
[13 := "Etat5" ;
VERS Fin_M_Com ;
ea9: ECRIRE ([23) := [15;
[13 := "Etat4";
VERS ea7;
ea5: [13 := "EtatER";
Fin D Com : VERS esyst1;

```

### CONCLUSION:

On a vu que notre système fonctionne comme un programme, dans lequel les dérouleurs (D\_COM et D\_CP) sont insérés, dont le temps de déroulement est strictement borné à chacune de leurs insertions. Car à la fin du quantum de travail qui leurs est affecté par le système, les dérouleurs, lui rendent la main et tout se passe comme s'ils s'interrompaient eux-mêmes au bon moment .

Pour que les choses marchent selon ce principe, il faut que les dérouleurs soient organisés en éléments finis bornés. C'est ce qui traduit dans la définition de l'autojectivité.<sup>5</sup>

Pour que les dérouleurs soient autojectifs, il faut que le déroulement de chaque chemin possible demande un temps borné et calculable à l'avance, et que les données traitées lors de chacune de ces circonstances occupent un volume dans la mémoire borné et aussi calculable à l'avance .

---

<sup>5</sup> Un dérouleur (procédure) autojectif est tel qu'il possède un point d'entrée unique et un point de sortie également unique, et tel que tout chemin qui joint le point d'entrée au point de sortie est fini borné.

### Contrôle du temps :

Le quantum de travail qu'on a défini pour notre étude, correspond au lancement, puis à la surveillance du déroulement d'une instruction d'une tâche en cours d'exécution. En effet dans nos langages utilisées (LCP, L.COM), il y a des instructions qui sont susceptibles de ne pas être finies bornées ce sont les boucles (REPETER SI ---- FIN\_REPET) et les conditions .

Ceci est contrôlé par le système, de manière que le dérouleur (D\_CP), lance la boucle et déroule par morceaux en décomposant en des éléments bornés tels que (REPETER SI Cond c1; inst1; ... instn; FIN\_REPET;), ce qui interdit le blocage du dérouleur par exemple sur une condition non vérifiée.

### Contrôle de l'espace :

Le système dispose de ses propres files, qui sont la file de la configuration, puis la file de déroulement<sup>6</sup> et la file de retour, qui sont finies-bornées et leurs volumes de mémoire est calculable à l'avance. A ces trois files on en ajoute deux autres, l'une pour le code du programme en LCP à dérouler, l'autre pour les variables de ce programme. A priori, ces deux files ont un volume fini-illimité. Dans ce cas, la solution adoptée est la suivante :

Au moment de la compilation d'un tel programme en LCP, on connaît déjà l'espace mémoire libre, en conséquence on n'en autorise l'implantation du code que lorsque son volume et l'espace à reperdre pour ses variables, ne dépasse pas l'espace restant.

---

<sup>6</sup> La file de déroulement contient au maximum 256 tâches à dérouler en parallèle à un instant donné, cette limite est strictement matérielle.



## DOUZZAVEDIBISAR

### LES AVENTURES D'EDITH CHEF DE GANG.

Quand Edith, les poings serrés arriva devant la villa de Michou le chef, elle était prête à tout. Le Parrain lui avait dit "Fonce !" Ce seul mot l'avait transformée en taureau furieux. La porte était cadénassée, elle la fit sauter d'un coup de pied. Dedans Michou surpris ... Hé, ho, plus que surpris, sidéré, la regardait comme si la Sainte Vierge venait de tomber du plafond. Tu imagines la Sainte Vierge ... Aussi hésita-t-il à défourailler son 49,3 . Cela lui fut fatal, Edith le cueillit d'un autre coup de pied dans les couilles, suivi d'un direct du talon aiguille en pleine poire. "Merde elle a un slip mauve ..." eut-il juste le temps de jeter avant qu'Edith ne le fit disparaître sur un: "Foutez-moi cette lope à la poubelle !" Que ses sbires s'empressèrent d'exécuter.

C'est comme si le tonnerre avait pétrifié la bande à Michou. Seul, l'Enflure, qui en avait vu d'autres, mais quand même ... les bras un peu écartés, la tête en avant, les yeux exorbités, il les avait déjà pas mal à l'état de repos, mais là c'était du spectacle, eut le temps de jeter un "Heu!" du creux de l'estomac.

"Maintenant les mecs vous allez en chier, et d'abord vous me dites Monsieur ... bien bien polis !"

Là, les mecs ils étaient sur le cul, façon de parler, car pour pas mal, le calebar il commençait à changer de couleur, et pas que politique. "Oui M'sieu " qu'ils tremblotèrent en chœur, la bande des caves. "Mieux que ça !" et après l'unisson d'un toussotement: "Ok meuh ... sieuh"

C'est comme ça que la bande du faubourg hexagone changea de Chef.

Le lendemain, sur le coup de onze heures, la Edith elle avait le sommeil calme et matutieux, elle se rapplique avec ses porte-flingues, inutile de casser la porte c'était fait, la voila comme une fleur. Les caves ils étaient toujours là, ils avaient pas changé de place, il leur avait bien fallu toute la nuit pour saisir le petit quelque chose de changé ... Hé ho, faut du temps pour comprendre. " M'sieu qu'est-ce qu'on va faire ?" qu'il dit l'Enflure, le premier ressaisi. "D'abord ta gueule, ensuite tu fais comme d'habitude; Toi, bretelles si t'ouvres ta grande gueule, t'as mon pied dans le cul ..."

Le "Ben ma salope ..." était pas encore sorti de sa bouche qu'il était étalé avec un gros gnou tout rouge sur sa joue gauche. "Et vous deux, là, les mi-portions vous surveillez bien les seaux qu'on vous a confiés, ils sont pleins de merde et pas de vagues ..." . Avec des jappements joyeux, Kiki et Nana vinrent lui lécher la chaussure gauche.

Là dessus tous ces pantins congelés reprennent vie car dès l'instant où on parle soupe .. et de s'agiter en tous sens en se frottant les mains. "Mais que degun bouge sans les ordres" qu'elle assaisonne l'Edith. "D'abord les problèmes

importants, je vois avec Gailuron la question de la concurrence, va falloir faire mieux que le Janmarri, OK ? Tu vas presto me virer un quelconque bougnoul ! Pour le ramassage du fric toi l'Enflure tu vas marnier et d'abord tu me sors la Poltax et vite si tu veux pas tâter de mon 49,3". "Putain ces gonzesses, ya plus moyen .." Qu'il dit l'Enflure entre ses dents juste pas assez fort pour qu'elle entende pas.

Manque de bol c'est par le journal du lendemain qu'elle apprit que c'était le Dabe lui-même qui avait viré le Diouri. Le Parrain l'avait doublée, ah ça !

Comment l'enflure réussit-il à découvrir le pot-aux-roses ? Le plus simplement du monde. Curieux comme une gonzesse, il se demandait pourquoi le chef s'isolait chaque jour pendant un bon moment dans la chambre forte. C'est lui qui l'avait fait construire, cette chambre, pour discuter au plus haut niveau des problèmes délicats de finances internationales, avec les autres chefs de gang, du temps de Michou, aussi était-il le seul à connaître le petit judas qu'on atteint par un petit passage discret... Ce soir là, installé devant son hublot invisible il vit. Le mobilier consistait en un simple et ample divan, un immense miroir sur pied qui lui tournait le dos, et un grand bureau de grand style, lequel, l'Enflure n'en savait trop rien, mais peu lui importait, il n'était pas là pour sa culture, du moins pas celle-là. Il sentit que quelqu'un s'agitait hors de son champ visuel. Le souffle lui manqua quand il vit bondir une Edith toute nue, affublée d'un justaucorps de cuir et de fer brillant dans ses clous et ses chaînes, le fouet à la main. Il était fasciné par la touffe fauve qui pointait le confluent nerveux de ses cuisses. Bien qu'un peu fripée de visage le reste semblait encore bien tenir la route... "Ben ma salope !" grinça l'Enflure entre ses dents. Il y avait l'image mais le son arrivait très atténué: la qualité de l'isolation, c'était pas de la technique HLM.

Et là commença une sorte de danse du scalp devant le miroir, sur un rythme bizarre ponctué par de violentes rotations de la tête qui projetait sa chevelure abondante dans tous les sens en lui fouettant le visage, ce qui semblait l'exciter davantage. Et puis sur un mouvement gracieux de ses seins qui bondissaient vers le ciel elle fouettait violemment le miroir en hurlant "morfle fumier en attendant que je te crève !". L'Enflure percevait distinctement ces fortes paroles et il en frissonnait, la scène commençait à lui faire un effet bizarre, il sentait se réveiller chez lui des forces qu'il croyait éteintes depuis longtemps. Puis dans un paroxysme, elle se jeta violemment à plat dos sur le divan et surcambrée, se mit à fourrager avec frénésie dans son silo à missiles. L'Enflure en bavait des ronds de chapeau, et lui-même commençait à s'agiter de la manchette.

Tout-à-coup, dans un deuxième paroxysme, et s'aidant du manche de son fouet, elle fit trois bonds dans une cambrure fantastique et se mit à pousser un hurlement dont la stridence réussit à percer le verre épais du hublot. L'Enflure sentit un frisson glacé lui monter du coccyx jusqu'à la cervicale et redescendre avec un effet de scie jusqu'à son fessier de gros porc. D'un coup de reins digne d'un acrobate elle fut debout face au miroir, et son fouet claqua derechef, le miroir déséquilibré par la violence, tourna un peu sur lui-même, et l'Enflure aperçut collée sur le miroir, l'image du dabe lacérée et déchiquetée. La virago, les lèvres retroussées jusqu'aux oreilles sur des dents de hyène, les mains crispées au bout de ses bras tendus poussa un dernier cri rauque "J'aurai ta peau vieux

pourri ... " Et elle s'effondra en position de grand écart, allongée sur sa jambe droite.

L'Enflure tout rêveur s'en alla en murmurant "Ayayaïe, encore un grand vizir qui veut être calife à la place du calife..." Le ton contenait un demi-doigt d'amertume, et deux doigts de regrets ...

E.B.

Toute ressemblance avec des gens ayant existé, existant ou devant exister un jour ne saurait être que fortuite et dépendante du plus grand des hasards, de toute manière faudrait être bien narre pour se conduire de la sorte.

