

⑫

DEMANDE DE BREVET D'INVENTION

A1

⑫② Date de dépôt : 17.01.92.

⑫③ Priorité :

⑫④ Date de la mise à disposition du public de la
demande : 23.07.93 Bulletin 93/29.

⑫⑤ Liste des documents cités dans le rapport de
recherche : *Le rapport de recherche n'a pas été
établi à la date de publication de la demande.*

⑫⑥ Références à d'autres documents nationaux
apparentés :

⑦① Demandeur(s) : BURNDY ELECTRA N.V. — BE.

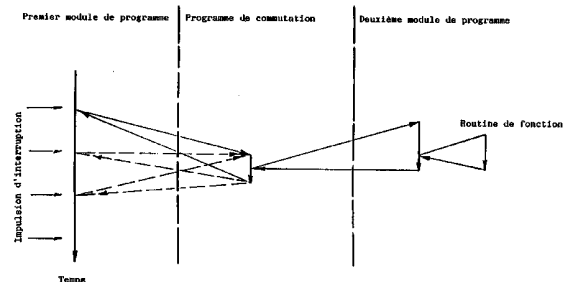
⑦② Inventeur(s) : Andries Luc.

⑦③ Titulaire(s) :

⑦④ Mandataire : Cabinet Plasseraud.

⑤④ Méthode pour la commande d'un système à l'aide d'un ordinateur.

⑤⑦ La méthode permet de commander un système à l'aide d'un ordinateur comportant un rythmeur qui envoie des impulsions d'interruption à une fréquence déterminée. Le programme de commande a trois modules. Le premier module assure la communication avec l'utilisateur via le clavier et le moniteur pour le traitement de données destinées au système ou en provenant et pour le stockage de données en mémoire et les autres fonctions non liées au système ou au temps. Le deuxième module contient des fonctions de commande du système aux instants où les commandes doivent être exécutées, ainsi qu'une routine pour l'exécution des commandes. Le troisième module est un programme de commutation du premier module au deuxième et vice-versa. Chaque impulsion d'interruption interrompt l'exécution du premier programme partiel et valide le programme de commutation qui commute vers le deuxième module pour l'exécution de la fonction de commande correspondant à l'instant concerné. Après l'exécution de la fonction de commande, le programme retourne à l'étape interrompue du premier module.



Procédé pour la commande d'un système à l'aide d'un ordinateur

L'invention concerne un procédé pour la commande d'un système à l'aide d'un ordinateur comportant un rythmeur qui envoie des impulsions d'interruption à une fréquence déterminée, ordinateur
5 pouvant fonctionner sous un système d'exploitation usuel comme MS-DOS.

Dans le procédé de ce genre connu, l'ordinateur est utilisé pour le traitement de données et éventuellement pour faire un nouveau programme de
10 commande, la commande réelle du procédé ayant lieu à l'aide de régulateurs PID séparés, d'un PLC (contrôleur logique programmable) ou d'un processeur en temps réel. Dans le cas de la commande par un processeur en temps
15 cyclique et ce programme de commande reste inchangé durant l'exploitation du système. Le PLC ou le processeur peut fournir à l'ordinateur des données qui sont traitées dans l'ordinateur et éventuellement
visualisées sur un écran.

20 Un procédé de ce genre présente l'inconvénient que le programme de commande ne peut pas être modifié durant l'exploitation du procédé. Le programme de commande est placé à l'endroit le moins flexible, c'est-à-dire dans les régulateurs PID ou dans
25 le langage de programmation le plus bas du PLC ou du

processeur en temps réel. L'utilisateur doit donc maîtriser plusieurs langages de programmation pour élaborer un programme de commande. De plus, les coûts de l'appareillage nécessaire pour l'application du procédé
5 connu sont relativement élevés, car il faut des régulateurs PID, un PLC ou un processeur en temps réel, ainsi qu'un ordinateur.

Jusqu'à présent, l'expert est d'avis qu'un ordinateur personnel ne convient pas pour la commande
10 directe d'un système, parce que le système d'exploitation habituel d'un ordinateur, comme le DOS, n'est pas un système d'exploitation en temps réel, alors que l'une des exigences les plus importantes pour un programme de commande est que la commande soit en temps
15 réel. Sous le système d'exploitation usuel d'un ordinateur personnel, les instructions d'un programme sont exécutées de manière séquentielle, et le temps nécessaire pour l'exécution d'instructions déterminées, comme les opérations avec un disque de mémoire, n'est
20 pas connu à l'avance. Le système d'exploitation attend pour passer à l'exécution d'une instruction que l'instruction précédente soit exécutée. De ce fait, il est impossible, avec le système d'exploitation usuel, de garantir une réponse de l'ordinateur dans un intervalle
25 de temps pré-déterminé ou de garantir qu'une fonction de commande sera exécutée à un instant pré-déterminé.

Dans ce contexte, on remarque que l'on entend par commande en temps réel que le programme de commande doit réagir dans un intervalle de temps pré-
30 déterminé et que le programme de commande doit pouvoir exécuter une fonction de commande à un moment pré-déterminé avec une résolution pré-déterminée.

De plus, il faut remarquer que l'on entend par système, dans la présente demande de brevet, un
35 procédé, une machine ou équivalent.

L'invention a pour but de créer un procédé pour la commande d'un système du genre cité au début dans lequel la commande peut avoir lieu uniquement à l'aide d'un ordinateur qui commande directement le
5 procédé via des unités usuelles d'interface, comme par exemple des convertisseurs numérique/analogique et des convertisseurs analogique/numérique et équivalents.

Pour cela, le procédé selon l'invention est caractérisé par un programme de commande qui comporte
10 trois modules de programme, un premier module de programme assurant la communication avec l'utilisateur via le clavier et le moniteur, pour le traitement de données destinées au système ou provenant du système et pour le stockage de données dans une mémoire et pour
15 toutes les autres fonctions non liées au temps ou au système, un deuxième module de programme comportant une série de fonctions de commande pour la commande du système avec des instants correspondants auxquels les fonctions de commande respectives doivent être
20 exécutées, ainsi qu'une routine pour l'exécution des fonctions de commande, et un troisième module de programme exécuté comme programme de commutation commutant du premier module de programme vers le deuxième et vice-versa, le rythmeur de l'ordinateur
25 étant programmé de telle manière que des impulsions d'interruption sont émises à une fréquence correspondant à une résolution de temps désirée pour la commande du système et chaque impulsion d'interruption interrompant l'exécution du premier module de programme
30 et mettant le programme de commutation en service, ce programme de commutation tenant une horloge en temps réel et commutant sur le deuxième module de programme pour l'exécution de la fonction de commande correspondant à l'instant concerné, après quoi on
35 revient à l'étape de programme interrompue du premier

module de programme.

Le procédé selon l'invention est basé sur le fait qu'en modifiant la fréquence des impulsions d'interruption fournies par le rythmeur usuel de l'ordinateur personnel, on obtient des impulsions de commande avec une fréquence telle qu'il est possible de satisfaire à une résolution de temps exigée pour la commande du système pour l'exécution d'une fonction de commande à un instant pré-déterminé. A chaque impulsion d'interruption, le premier module de programme est interrompu et le programme de commutation qui tient une horloge en temps réel est mis en action, après quoi le deuxième module de programme est parcouru pour l'exécution d'une fonction de commande correspondant à l'instant concerné. On obtient ainsi que l'ordinateur commande le système en temps réel, un seul langage de programmation étant nécessaire pour réaliser le programme de commande. Ce programme de commande est bien accessible et peut au besoin être modifié et adapté via le premier module de programme durant l'exploitation du système. Le coût de l'appareillage nécessaire pour l'exécution du procédé est relativement bas.

Selon une forme d'exécution préférée de l'invention, les fonctions de commande du deuxième module de programme sont reprises dans un tableau divisé en une série de fenêtres de temps égales successives contenant chacune une fonction de commande, le deuxième module de programme comportant une routine d'interprétation qui est parcourue après chaque commutation du premier au deuxième module de programme et qui tient un registre d'index pour la détermination d'une fenêtre suivante à appeler puis lit la fonction mémorisée dans la fenêtre, après quoi cette fonction est exécutée, et, après l'exécution de cette fonction de commande, la routine d'interprétation retourne au

programme de commutation qui commute sur l'étape interrompue du premier module de programme. Cela rend l'adaptation du programme de commande relativement simple, car il suffit d'apporter les modifications dans
5 le tableau des fonctions de commande.

De préférence, le tableau des fonctions de commande contient des numéros de fonction et des adresses d'argument correspondantes, les numéros de fonctions renvoyant à une bibliothèque de fonctions de
10 commande et les adresses d'arguments aux emplacements de mémoire d'une mémoire d'arguments, la routine d'interprétation faisant exécuter la fonction de commande correspondant au numéro de fonction lu avec les arguments de l'adresse d'arguments correspondante. De
15 cette manière, il suffit pour modifier une fonction de commande de modifier le numéro de fonction et l'adresse d'argument dans une fenêtre de temps, et aucune modification dans la bibliothèque des fonctions de commande n'est nécessaire.

20 Selon une forme d'exécution avantageuse de l'invention, chaque fonction de commande se compose d'une routine de fonction avec un certain nombre d'étapes où, dans la première étape, l'argument est lu dans la mémoire d'arguments, et dans l'étape suivante
25 une commande d'entrée/sortie vers le système à commander peut être exécutée, après quoi un traitement de données est effectué, puis la routine de fonction retourne à la routine d'interprétation.

L'invention est expliquée plus en détail ci-dessous à l'aide du dessin, dans lequel quelques schémas fonctionnels illustrant le procédé selon l'invention
30 sont donnés.

La fig. 1 est un schéma fonctionnel du premier module de programme d'une forme d'exécution du
35 procédé selon l'invention.

La fig. 2 est un schéma fonctionnel du programme de commutation correspondant à la fig. 1.

La fig. 3 est un schéma fonctionnel du deuxième module de programme correspondant à la fig. 1.

5 La fig. 4 est un schéma fonctionnel d'une routine de fonction correspondant à la fig. 3.

La fig. 5 montre de manière schématique le saut entre les différents programmes du programme de commande.

10 La fig. 6 représente schématiquement l'accès du premier et du deuxième module de programme à quelques mémoires communes.

Dans la fig. 1, on voit sous la forme d'un schéma fonctionnel fortement simplifié le premier module
15 d'un programme de commande pour la commande d'un système à l'aide d'un ordinateur personnel. Ce premier module tourne de la manière habituelle sous le système d'exploitation standard de l'ordinateur, comme par exemple MS-DOS, et le programme contrôle dans chaque
20 cycle si des données sont introduites via le clavier, après quoi une autre des étapes de programme indiquées est exécutées. Ce module assure ainsi la communication entre le système à commander et l'utilisateur.

Comme on le voit à la fig. 1, une première
25 étape de programme regarde s'il y a eu une entrée via le clavier, après quoi une étape de traitement de données est parcourue dans laquelle des calculs sont effectués avec les données introduites via le clavier ou peuvent émaner du système commandé, comme on le verra ci-
30 dessous. Si nécessaire, l'information reproduite sur un moniteur usuel de l'ordinateur est mise à jour. Ensuite on peut parcourir une étape de programme dans laquelle les données traitées et éventuellement d'autres données provenant du système commandé sont emmagasinées dans une
35 mémoire, par exemple un disque dur.

Enfin, ce premier module comporte une routine d'adaptation au moyen de laquelle on peut apporter des modifications dans un tableau de fonctions de commande, comme on l'expliquera plus en détail ci-dessous.

Comme ce premier module tourne sous la commande du système d'exploitation standard de l'ordinateur par lequel l'exécution des différentes étapes de programme ne peuvent pas avoir lieu à des instants pré-déterminés, ce module peut également être nommé module asynchrone.

L'ordinateur personnel à utiliser est évidemment équipé d'un rythmeur qui, dans le cas du système d'exploitation MS-DOS ou PC-DOS, fournit des impulsions d'interruption à une fréquence d'environ 18,2 Hz. A chaque impulsion d'interruption, une routine DOS standard est parcourue, dans laquelle les fonctions suivantes sont exécutées:

- la mise à jour de l'horloge DOS;
- l'indication d'un changement de date;
- si nécessaire la mise à l'arrêt du moteur de l'unité à disquette.

La fréquence à laquelle le rythmeur émet des impulsions d'interruption peut être changée par la programmation. Selon l'invention, cette fréquence est choisie telle qu'il est satisfait à la résolution de temps voulue pour l'exécution des différentes fonctions de commande. Comme la fig. 5 le montre schématiquement, l'exécution du premier module de programme est chaque fois interrompue par une impulsion d'interruption du rythmeur et un programme de commutation, que l'on peut désigner par "timer interrupt handler" peut être indiqué, ce programme de commutation remplaçant la routine DOS standard.

Dans la fig. 2 on voit un schéma fonctionnel

fortement simplifié de ce programme de commutation.
Comme on peut le voir à la fig. 2, chaque impulsion
d'interruption est suivie du déroulement d'une première
étape de programme dans laquelle les fonctions de la
5 routine DOS remplacée sont exécutées, de sorte que ces
fonctions sont exécutées à la fréquence originelle de
cette routine DOS. Après cette étape de programme, une
horloge en temps réel est mise à jour puis il y a une
commutation vers un deuxième module de programme, comme
10 indiqué schématiquement à la fig. 5. Après que le
deuxième module de programme, dont un schéma fonctionnel
fortement simplifié est donné à la fig. 3, a été
parcouru, on retourne via le programme de commutation à
l'endroit où le premier module de programme avait été
15 interrompu. La tâche du programme de commutation est
donc de commuter à des instants fixes du premier module
sur le deuxième module. Dans ce deuxième module, une
fonction de commande est exécutée. Cela signifie qu'une
fonction de commande peut être exécutée chaque fois à un
20 moment pré-déterminé par le rythmeur. Comme le deuxième
programme peut être parcouru à des instants fixes, il
peut être considéré comme programme synchrone.

Le deuxième module comporte une routine qui
peut être nommée routine d'interprétation en temps réel
25 et qui est parcourue après la commutation sur le
deuxième module de programme. Le deuxième module de
programme comporte également un tableau des fonctions de
commande qui est divisé en n fenêtres de temps
successives de même longueur. Dans chaque fenêtre de
30 temps, une fonction de commande et une adresse
d'argument peuvent être fixés. Comme le montre la fig.
3, dans la première étape de la routine d'interprétation
est tenu un registre d'index dont le contenu renvoie à
la fenêtre suivante à exécuter du tableau des fonctions
35 de commande. L'exécution d'une fonction de commande a

donc lieu à un moment déterminé par le numéro de la
fenêtre de temps correspondante. Ensuite la routine
d'interprétation lit le contenu de la fenêtre du
tableau des fonctions de commande indiquée par le
5 registre d'index. Comme on l'a déjà remarqué ci-dessus,
ce contenu consiste en un numéro de fonction et une
adresse d'argument qui renvoie respectivement à une
bibliothèque de fonction et à une adresse d'argument.
Ensuite la routine d'interprétation libère la fenêtre
10 lue, de sorte que cette fenêtre devient libre pour
l'introduction d'un nouveau numéro de fonction et d'une
nouvelle adresse d'argument pour le moment qui vient
après un cycle complet de parcours du tableau des
fonctions de commande. Ensuite la routine
15 d'interprétation saute sur la fonction de commande
indiquée par le numéro de fonction, qui consiste en une
fonction de routine dont la constitution générale est
donnée à la fig. 4.

Comme on peut le voir à la fig. 4, la
20 première étape de programme de la fonction de routine
consiste à prendre l'argument à l'emplacement de mémoire
d'argument indiqué par l'adresse d'argument. Selon le
genre de fonction de commande, la routine de fonction
exécute ensuite une commande d'entrée/sortie vers le
25 système commandé. Dans une étape de programme suivante,
la routine de fonction traite des données obtenues par
l'exécution de la commande d'entrée/sortie ou de
l'argument lu ou provenant d'une mémoire de données
commune au premier et au deuxième module de programme.

30 La routine de fonction peut de plus
comporter une routine d'adaptation permettant
d'introduire une adaptation dans le tableau des
fonctions de commande et à l'aide de laquelle une
nouvelle fonction dans le tableau des fonctions de
35 commande peut être reprise dans une fenêtre de temps

correspondant au moment où la nouvelle fonction doit être exécutée, et permettant de fixer un numéro de fonction et une adresse d'argument et d'écrire l'argument à l'adresse correspondante dans la mémoire d'arguments. De cette manière, la fonction exécutée peut modifier de manière dynamique le déroulement de la commande et le programme de commande peut adapter la commande du procédé aux effets des opérations de commande exécutées. On obtient ainsi un programme de commande dynamique apprenant par lui-même.

L'argument de la fonction de commande peut par exemple contenir également une date et une heure auxquelles cette fonction, par exemple le branchement d'un élément chauffant, doit être exécutée. Lorsque la comparaison de cette date et heure désirées avec l'horloge en temps réel indique que ce moment n'est pas encore arrivé, la fonction de commande peut se remettre, au moyen de la routine d'adaptation, dans une fenêtre de temps libre.

Comme on l'a déjà dit plus haut, le premier module de programme contient la même routine d'adaptation. L'utilisateur peut donc apporter des modifications en mode interactif dans le système de commande durant l'exploitation du système.

Il est possible que la routine d'adaptation ne puisse plus trouver de fenêtre de temps libre en cherchant une fenêtre de temps libre au moment désiré pour l'exécution de la nouvelle fonction de commande. Dans ce cas, la routine d'adaptation examine, à l'aide d'un schéma de priorités dans lequel les priorités pour les différentes fonctions de commande sont enregistrées, si la nouvelle fonction de commande ou la fonction de commande déjà mémorisée dans la fenêtre de temps doit être fixée dans une autre fenêtre de temps libre dont le moment correspondant se rapproche autant que possible du

moment désiré. Lors de la recherche d'une autre fenêtre libre, d'autres déplacements de fonctions de commande au sein du tableau des fonctions de commande sont possibles en fonction du schéma de priorités.

5 Lorsque la routine de fonction a libéré la place dans la mémoire d'arguments avec l'adresse d'argument correspondante, le programme revient à la routine d'interprétation et de là au programme de commutation, puis le programme de commutation revient au
10 premier module de programme, à l'endroit où ce module avait été interrompu par l'impulsion d'interruption.

Il est souhaitable que le temps du cycle de parcours du programme de commutation, du deuxième module de programme et de la routine de fonction soit plus
15 court que l'intervalle de temps entre deux impulsions d'interruption. Il est toutefois également possible d'affecter plus d'une fenêtre de temps à une fonction de commande à exécuter.

Avec le programme de commande de
20 l'invention, il est possible d'utiliser un ordinateur personnel usuel pour la commande en temps réel d'un système, car à chaque impulsion d'interruption, et donc à des instants déterminés, une fonction de commande peut chaque fois être exécutée.

25 Comme la fig. 6 le montre de manière très schématique, le premier module de programme et le deuxième module de programme possèdent quelques mémoires communes, ce qui permet à l'utilisateur d'une part de recevoir via le premier module des données sur le
30 procédé commandé, et d'autre part d'entrer des modifications dans le programme de commande durant l'exploitation. Comme on le voit à la fig. 6, les deux modules de programme ont accès au tableau des fonctions de commande, à la mémoire d'argument et à une mémoire de
35 données.

Bien que dans l'exemple d'exécution on indique que la fenêtre de temps et l'adresse d'argument correspondante sont chaque fois libérées après l'exécution de la fonction de commande concernée, il est
5 également possible de parcourir une série fixe de fonctions de commandes sans chaque fois apporter des modifications.

L'invention n'est pas limitée à l'exemple d'exécution décrit ci-dessus, qui peut varier de
10 différentes manières dans le cadre de l'invention.

REVENDEICATIONS

1. Procédé pour la commande d'un système à l'aide d'un ordinateur comportant un rythmeur qui envoie des impulsions d'interruption à une fréquence

5 déterminée, ordinateur pouvant fonctionner sous un système d'exploitation usuel comme MS-DOS, caractérisé par un programme de commande qui comporte trois modules programme, un premier module de programme assurant la communication avec l'utilisateur via le clavier et le

10 moniteur, pour le traitement de données destinées au système ou provenant du système et pour le stockage de données dans une mémoire et pour toutes les autres fonctions non liées au temps ou au système, un deuxième module de programme comportant une série de fonctions de

15 commande pour la commande du système avec des instants correspondants auxquels les fonctions de commande respectives doivent être exécutées, ainsi qu'une routine pour l'exécution des fonctions de commande, et un

20 commutation commutant du premier module de programme vers le deuxième et vice-versa, le rythmeur de l'ordinateur étant programmé de telle manière que des impulsions d'interruption sont émises à une fréquence correspondant à une résolution de temps désirée pour la

25 commande du système et chaque impulsion d'interruption interrompant l'exécution du premier module de programme et mettant le programme de commutation en service, ce programme de commutation tenant une horloge en temps réel et commutant sur le deuxième module de programme

30 pour l'exécution de la fonction de commande correspondant à l'instant concerné, après quoi on revient à l'étape de programme interrompue du premier module de programme.

2. Procédé selon la revendication 1,

35 caractérisé en ce que les fonctions de commande du

deuxième module de programme sont reprises dans un tableau divisé en une série de fenêtres de temps égales successives contenant chacune une fonction de commande, le deuxième module de programme comportant une routine d'interprétation qui est parcourue après chaque
5 commutation du premier au deuxième module de programme et qui tient un registre d'index pour la détermination d'une fenêtre suivante à appeler puis lit la fonction mémorisée dans la fenêtre, après quoi cette fonction
10 est exécutée, et, après l'exécution de cette fonction de commande, la routine d'interprétation retourne au programme de commutation qui commute sur l'étape interrompue du premier module de programme.

3. Procédé selon la revendication 2,
15 caractérisé en ce que le tableau des fonctions de commande contient des numéros de fonction et des adresses d'argument correspondantes, les numéros de fonctions renvoyant à une bibliothèque de fonctions de commande et les adresses d'arguments aux emplacements de
20 mémoire d'une mémoire d'arguments, la routine d'interprétation faisant exécuter la fonction de commande correspondant au numéro de fonction lu avec les arguments de l'adresse d'arguments correspondante.

4. Procédé selon la revendication 3,
25 caractérisé en ce que chaque fonction de commande se compose d'une routine de fonction avec un certain nombre d'étapes, où, dans la première étape, l'argument est lu dans la mémoire d'arguments, et dans l'étape suivante une commande d'entrée/sortie vers le système à commander
30 peut être exécutée, après quoi un traitement de données est effectué, puis la routine de fonction retourne à la routine d'interprétation.

5. Procédé selon les revendications 3 et 4,
caractérisé en ce que la dite routine d'interprétation
35 du deuxième module de programme libère, après l'appel de

la fenêtre de temps concernée, le contenu de cette fenêtre de temps pour une nouvelle fonction de commande, tandis que la routine de fonction libère l'adresse d'argument pour un nouvel argument.

5 6. Procédé selon la revendication 6, caractérisé en ce qu'au moins une fonction de commande contient une routine d'adaptation qui introduit une modification dans le tableau des fonctions de commande en mémorisant un nouveau numéro de fonction et une
10 nouvelle adresse d'argument en correspondance avec l'instant auquel la nouvelle fonction de commande doit être exécutée, un nouvel argument étant introduit à l'adresse d'argument dans la mémoire d'arguments.

 7. Procédé selon la revendication 5 ou 6, caractérisé en ce que le premier module de programme
15 contient une routine d'adaptation au moyen de laquelle l'utilisateur peut introduire une modification dans le tableau des fonctions de commande en entrant un nouveau numéro de fonction et une nouvelle adresse d'argument
20 dans une fenêtre de temps libre du tableau, en correspondance avec l'instant auquel la nouvelle fonction de commande doit être exécutée, un nouvel argument étant introduit à l'adresse d'argument dans la mémoire d'arguments.

25 8. Procédé selon la revendication 4, 5, 6 ou 7, caractérisé en ce que la recherche d'une fenêtre de temps libre pour l'introduction d'une modification dans le tableau des fonctions de commande est exécutée à l'aide d'un schéma de priorités dans lequel les
30 priorités des différentes fonctions de commande sont mémorisées.

 9. Procédé selon l'une des revendications ci-dessus, caractérisé en ce que les premier et deuxième modules de programme peuvent échanger des
35 données via une mémoire commune.

10. Procédé selon l'une des revendications ci-dessus, caractérisé en ce que le programme de commutation exécute les fonctions usuelles du système d'exploitation avec la fréquence originelle.

1/5

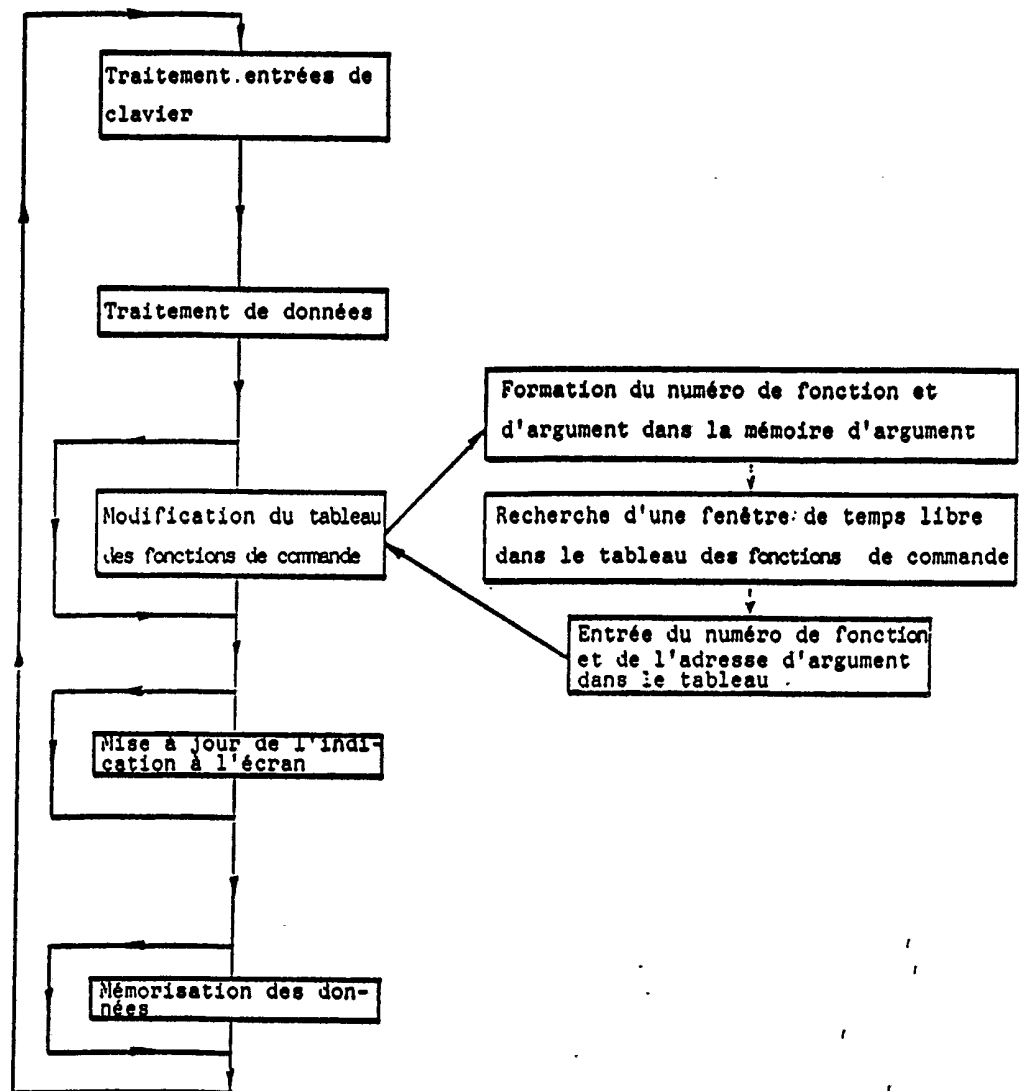


fig.1

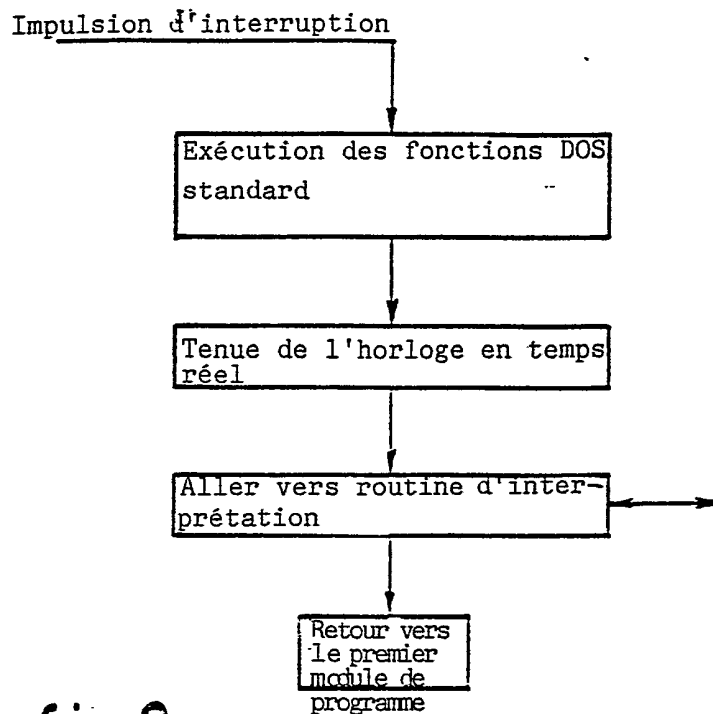


fig.2

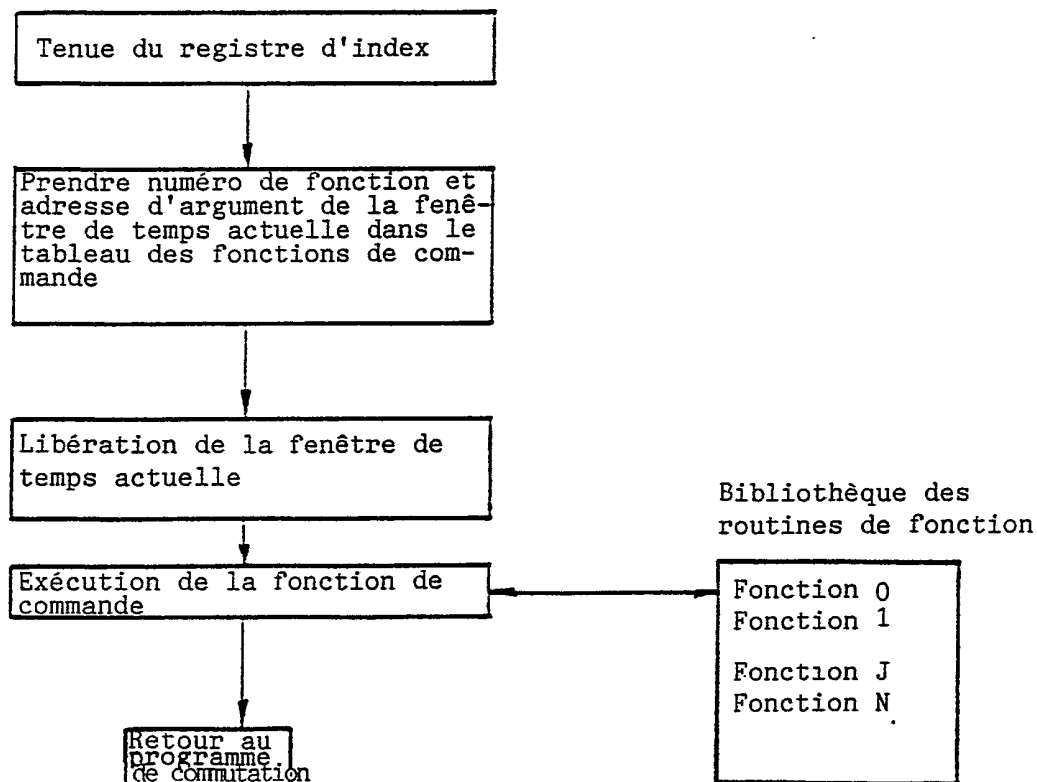


fig.3

3/5

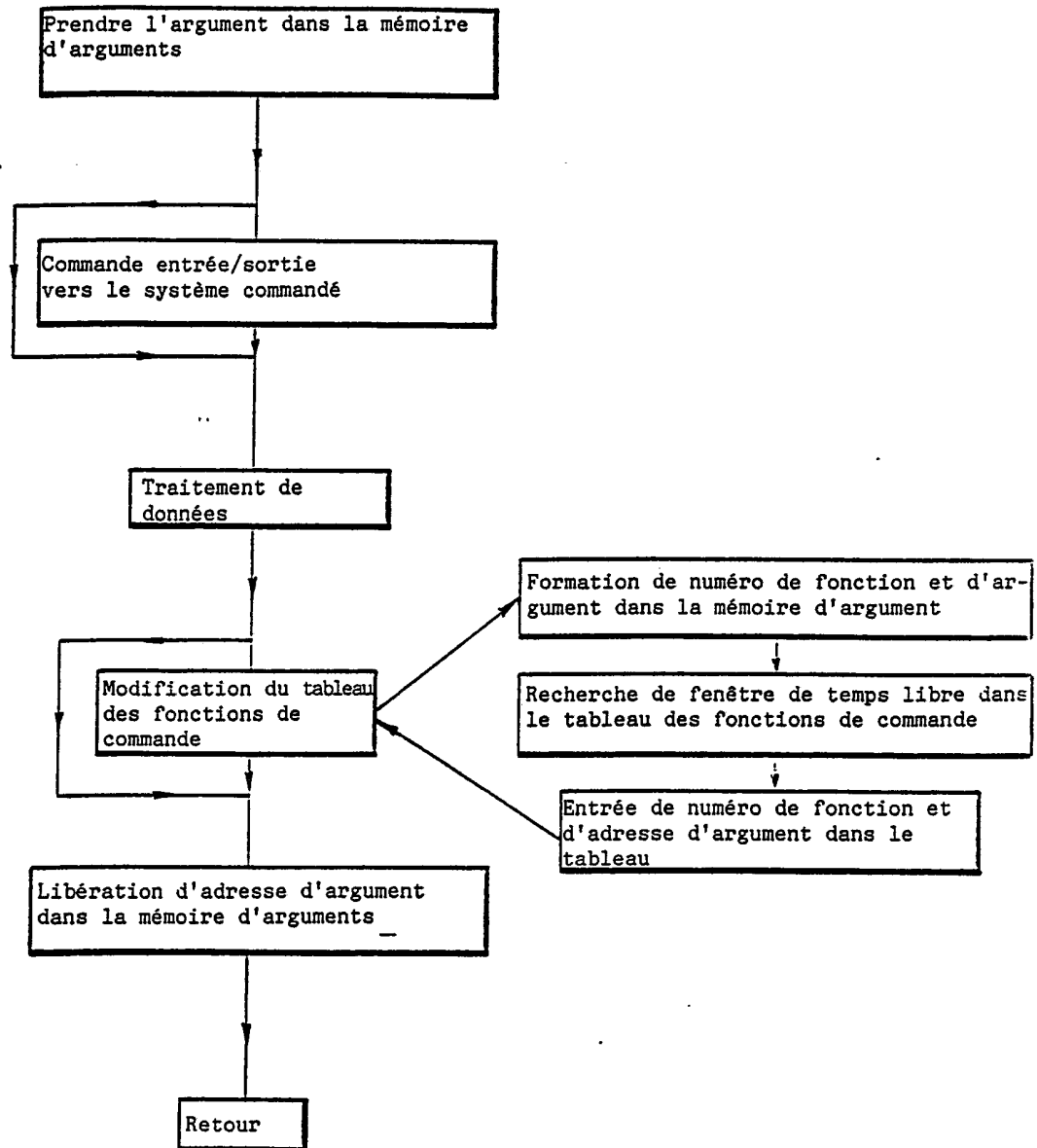


fig.4

4/5

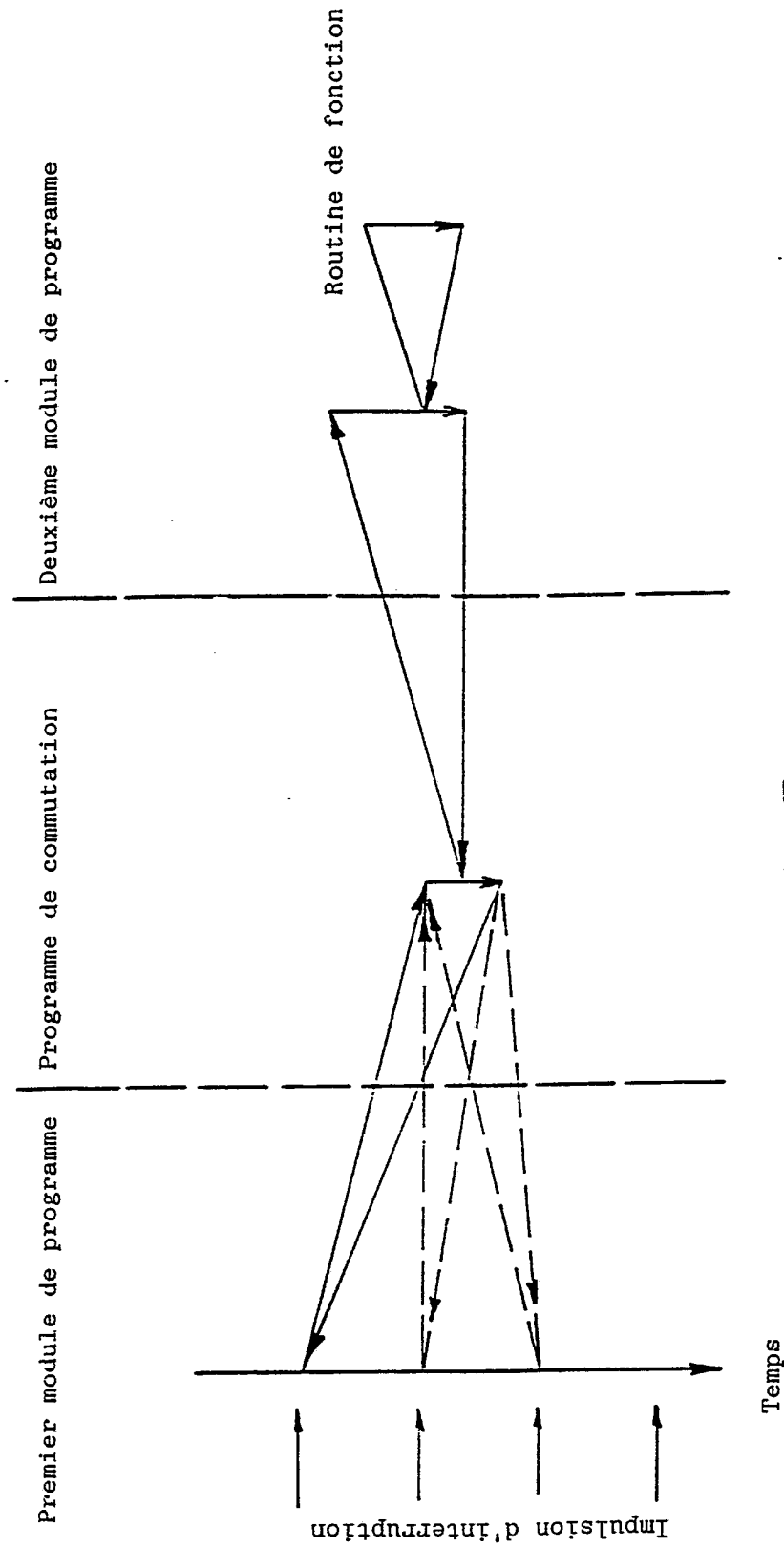


fig.5

5/5

Premier module
de programme

Mémoire

Deuxième module
de programme

Tableau des fonctions de commande

0	n° de fonction	adresse d'arg.
1	n° de fonction	adresse d'arg.
.	n° de fonction	adresse d'arg.
.	n° de fonction	adresse d'arg.
j	n° de fonction	adresse d'arg.
.	n° de fonction	adresse d'arg.
.	n° de fonction	adresse d'arg.
n	n° de fonction	adresse d'arg.

Mémoire d'arguments

Mémoire de données

fig.6