



- (51) **Classification internationale des brevets :**
G06F 9/48 (2006.01) **G06F 12/08** (2006.01)

(21) **Numéro de la demande internationale :**
PCT/EP2011/052833

(22) **Date de dépôt international :**
25 février 2011 (25.02.2011)

(25) **Langue de dépôt :** français

(26) **Langue de publication :** français

(30) **Données relatives à la priorité :**
10/51428 1 mars 2010 (01.03.2010) FR

(71) **Déposant (pour tous les États désignés sauf US) :**
SAGEM DEFENSE SECURITE [FR/FR]; 27 rue
Leblanc, F-75015 Paris (FR).

(72) **Inventeurs; et**

(75) **Inventeurs/Déposants (pour US seulement) :** **DUFOUR,**
Jean-Louis [FR/FR]; Sagem Défense Sécurité, 27 rue
Leblanc, F-75015 Paris (FR). **DESTELLE, Michel** [FR/
FR]; Sagem Défense Sécurité, 27 rue Leblanc, F-75015
Paris (FR).

(74) **Mandataire :** **MAILLET, Alain;** 5 place Newquay, B.P
70250, F-35802 Dinard Cedex (FR).

(81) **États désignés (sauf indication contraire, pour tout titre
de protection nationale disponible) :** AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO,
DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP,
KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD,
ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI,
NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD,
SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR,
TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **États désignés (sauf indication contraire, pour tout titre
de protection régionale disponible) :** ARIPO (BW, GH,
GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG,
ZM, ZW), eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ,
TM), européen (AL, AT, BE, BG, CH, CY, CZ, DE, DK,
EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,
LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, ML, MR, NE, SN, TD, TG).

Publiée :
— avec rapport de recherche internationale (Art. 21(3))

Publiée :

— avec rapport de recherche internationale (Art. 21(3))

(54) Title : DETERMINISTIC MULTI-TASK SEQUENCING METHOD

(54) Titre : PROCÉDÉ DE SÉQUENCÉMENT DÉTERMINISTE MULTITÂCHE

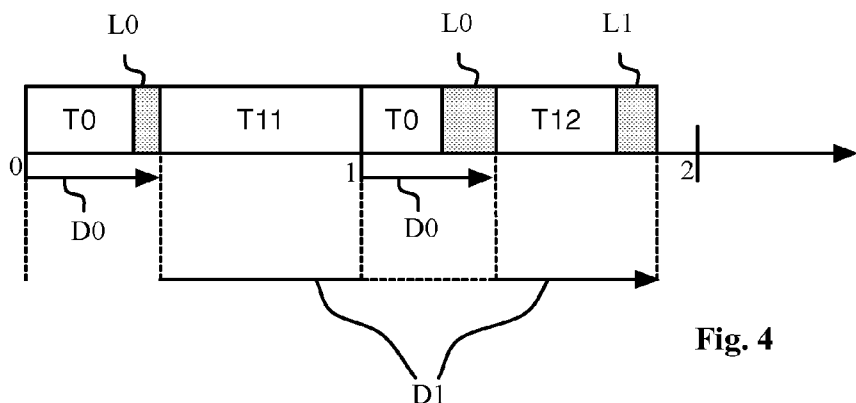


Fig. 4

(57) Abstract : The invention relates to methods enabling predictability of the run times for systems used in environments where security is critical. The invention likewise relates to the influence of the cache memory on the operation of such sequencing methods. The task sequencing method is described in a preemptive multi-task system where the launch instants of the tasks are determined. Said method advantageously invalidates the cache memory when switching between tasks but enables the write-back operating mode of said cache memory under certain conditions. Said method enables the run time of each task to be made separate while at the same time maintaining a high level of performance.

(57) Abrégé : On s'intéresse aux procédés permettant une prédictibilité des durées d'exécution destinés aux systèmes utilisés dans des environnements où la sécurité est critique. On s'intéresse également à l'influence de la mémoire cache sur le fonctionnement de tels procédés de séquençement. Le procédé de séquençement des tâches est décrit dans un système multitâche préemptif où les moments de lancement des tâches sont déterminés. Avantagusement, ce procédé invalide la mémoire cache lors du basculement entre tâches mais permet toutefois le mode de fonctionnement write-back de cette mémoire cache selon certaines conditions. Ce procédé permet de rendre indépendante la durée d'exécution de chaque tâche tout en maintenant un niveau élevé de performance.

Procédé de séquençement déterministe multitâche

La présente invention concerne le domaine des procédés de séquençement des tâches dans un système d'information multitâche. Plus particulièrement, on s'intéresse aux procédés permettant une prédictibilité des durées d'exécution destinés aux systèmes utilisés dans des environnements où la sécurité est critique. De tels environnements sont par exemple le domaine des logiciels embarqués dans les avions, les centrales nucléaires ou les systèmes militaires. On s'intéresse également à l'influence de la mémoire cache sur le fonctionnement de tels procédés de séquençement.

Le fonctionnement d'un processeur peut se schématiser en cycles consistant en une première étape de transfert d'une instruction et de données de la mémoire vers des registres du processeur, une seconde étape exécutant l'instruction sur ces registres et une troisième étape transférant le résultat des registres vers la mémoire.

On appelle tâche, une fonction logicielle s'exécutant sur un processeur et typiquement constituée d'une succession de tels cycles de calcul.

Les mémoires cache sont des blocs de mémoire rapide servant d'intermédiaire entre le processeur et la mémoire principale contenant les instructions et les données.

La mémoire cache se caractérise par un temps d'accès d'un ou de deux ordres de grandeur inférieurs à un accès à la mémoire principale. Typiquement, au démarrage d'une tâche, cette mémoire cache est vide et les accès mémoire s'effectuent en mémoire principale. Les instructions et données lues en mémoire principale restent
5 mémorisées dans la mémoire cache. Un nouvel accès à ces instructions ou données provoque leur lecture dans la mémoire cache, donc beaucoup plus rapide, et non dans la mémoire principale. Des processus connus sous les noms de *write-through* et *write-back* sont deux politiques bien connues de gestion de la cohérence entre le contenu de la mémoire cache et celui de la mémoire principale. Ces politiques de cohérences
10 seront détaillées par la suite. On comprend aisément que l'utilisation de mémoire cache influence grandement la durée d'exécution d'une tâche donnée.

Lorsque l'on désire effectuer plusieurs tâches de manière concurrente sur un même processeur, ces tâches doivent être exécutées par segments successifs avec des basculements entre chaque segment. En effet, un processeur est fondamentalement
15 mono tâche. Pour gérer cette exécution entrelacée, il faut mettre en œuvre un séquenceur qui, lorsqu'il est lancé, décide, en fonction de sa politique de séquençement, s'il interrompt la tâche courante et dans ce cas quelle tâche parmi celles qui sont en attente va devenir la tâche courante. Chaque basculement entraîne un changement de contexte où le compteur ordinal, le pointeur sur l'instruction
20 suivante à exécuter, et les registres du processeur sont sauvegardés avant le basculement pour être restaurés lorsque la tâche redevient active. On distingue plusieurs modes de fonctionnement multitâche. Historiquement, le premier mode apparu est le mode de fonctionnement coopératif. L'appel au séquenceur s'effectue à l'occasion des appels au système, en particulier lors de la fin de tâche, et doit donc
25 être prévu dans le code de chaque tâche. Ce mode possède l'inconvénient qu'il n'est pas possible d'assurer l'équité de l'accès au processeur pour les différentes tâches. Un problème logiciel provoquant une boucle infinie, par exemple, interdit l'accès aux autres tâches.

Un second mode de fonctionnement multitâche consiste en un mode dit
30 préemptif. Dans ce mode de fonctionnement, le séquenceur est lancé également sur occurrence d'événement(s) extérieur(s) à la tâche courante. Un séquenceur préemptif périodique provoque le basculement d'une tâche active vers une autre tâche à l'échéance de chaque période de séquençement et à la fin de l'exécution d'une instance d'une tâche. Il est alors possible, lorsque c'est désiré, d'assurer l'équité

d'accès au processeur ou d'implémenter toute politique de séquençement souhaitée. Celle-ci peut tenir compte de la disponibilité d'autres ressources ou de niveau de priorité entre les tâches. C'est le mode de fonctionnement des systèmes multitâches les plus répandus actuellement.

5 Le moment du basculement entre deux tâches lors du fonctionnement d'un séquenceur multitâche préemptif influe sur la durée totale d'exécution de la tâche du fait de l'utilisation de la mémoire cache. En effet, lors de l'exécution d'une tâche, le taux de remplissage en données utiles de la mémoire cache évolue. Lors du basculement, les données utiles à une tâche sont généralement inutiles à la nouvelle.

10 Si ce basculement a lieu à un moment où le cache est presque vide en données utiles, elle est peu pénalisante. Tandis que lorsque le basculement a lieu alors que la mémoire cache est remplie de données utiles, elle est pénalisante, car on perd ces données utiles qui devront être rechargées depuis la mémoire principale lorsque la tâche redevient active.

15 Dans un contexte où le système d'information doit répondre à des contraintes de sécurité critiques, il est demandé de garantir un majorant sur la durée d'exécution de chaque tâche. Dans ce contexte, il est avantageux de garantir à chaque tâche un temps d'exécution qui ne dépend que de la tâche considérée et n'est pas influencé par le déroulement des autres tâches.

20 L'invention vise à atteindre l'objectif d'indépendance précédent en proposant un procédé de séquençement des tâches dans un système multitâche préemptif où les moments de lancement des tâches sont déterminés. Avantageusement, ce procédé invalide la mémoire cache lors du basculement entre tâches mais permet toutefois le mode de fonctionnement retardé (*write-back* en anglais) de cette mémoire cache selon

25 certaines conditions. Ce procédé permet de rendre indépendant la durée d'exécution de chaque tâche tout en maintenant un niveau élevé de performance.

 L'invention concerne un procédé de séquençement d'un ensemble de tâches périodiques sur un processeur de traitement de l'information, chaque tâche ayant une période propre, le séquençement étant périodique et préemptif, signifiant que le

30 séquençement provoque le basculement d'une tâche active vers une autre tâche à l'échéance de chaque période de séquençement et à la fin de l'exécution d'une instance d'une tâche, qui comporte les étapes suivantes :

- pour chaque tâche, la détermination d'une limite de durée allouée à l'exécution d'une instance de cette tâche ;

- lorsque l'exécution d'une instance d'une tâche s'achève est introduite une étape d'attente que la limite de durée allouée à ladite tâche s'écoule avant de provoquer le basculement.

- lors d'un basculement, la mémoire cache est invalidée.

5 Selon un mode particulier de réalisation de l'invention, le processeur possédant une mémoire cache pouvant fonctionner dans un mode transparent et dans un mode retardé, le procédé comporte en outre lors du basculement d'une tâche active vers une nouvelle tâche :

10 - une étape consistant à forcer le mode transparent de la nouvelle tâche active si ce n'est pas son segment final.

 Selon un mode particulier de réalisation de l'invention, le procédé comprend en outre :

15 - lors de la préemption d'une tâche en mode retardé vers une nouvelle tâche, une étape de mise en cohérence du cache en un temps constant, avant l'étape d'invalidation ; et

 - sur des segments non-finaux choisis, l'omission de l'étape consistant à forcer le mode transparent de la nouvelle tâche active.

20 L'invention concerne également un dispositif de traitement de l'information qui comporte au moins un processeur capable d'exécuter des tâches ; un moyen de séquençement périodique et préemptif signifiant que le séquençement provoque le basculement d'une tâche active vers une autre tâche à l'échéance de chaque période de séquençement et à la fin de l'exécution d'une instance d'une tâche, pour le séquençement d'un ensemble de tâches périodiques, chaque tâche ayant une période propre ; et qui dispose en outre de moyens de prise en compte pour chaque tâche

25 d'une limite de durée allouée à l'exécution d'une instance de cette tâche ; de moyens d'introduire une étape d'attente que la limite de durée allouée à une tâche s'écoule avant de provoquer le basculement lorsqu'une instance de ladite tâche s'achève et de moyens d'invalider la mémoire cache lors d'un basculement.

30 Les caractéristiques de l'invention mentionnées ci-dessus, ainsi que d'autres, apparaîtront plus clairement à la lecture de la description suivante d'un exemple de réalisation, ladite description étant faite en relation avec les dessins joints, parmi lesquels :

 La Fig. 1 illustre un exemple de découpage de tâche à la conception dans un séquenceur périodique non-préemptif,

La Fig. 2 illustre le même exemple de découpage de tâche à l'exécution dans un séquenceur périodique préemptif,

La Fig. 3 illustre l'influence sur les autres tâches que peut avoir un vol de temps par une tâche,

5 La Fig. 4 illustre le fonctionnement du séquenceur selon un exemple de réalisation de l'invention.

L'invention se place dans le cadre de l'exécution périodique de tâches sur un processeur dans un système de traitement de l'information. On cherche à pouvoir garantir le fonctionnement de chaque tâche et à en maîtriser la consommation en
10 temps processeur. Pour cela, il faut déterminer une durée maximale de fonctionnement de chaque tâche selon un scénario de pire cas. Quand on a fait cela, la sécurité vient de mécanismes empêchant qu'un dysfonctionnement d'une tâche puisse se répercuter sur le fonctionnement des autres tâches.

Pour illustrer la plus-value d'un séquençement préemptif, un exemple de
15 découpage de tâche dans un séquenceur périodique non-préemptif est donné Fig. 1. Sur cette figure, la période est marquée par des tirets verticaux numérotés de 0 à 6. Dans cet exemple, nous avons deux tâches T0 et T1, toutes deux périodiques, qui doivent s'exécuter. La tâche T0 a une durée inférieure à la durée de la période et doit s'exécuter toutes les périodes. La tâche T1 a une durée supérieure à la durée de la
20 période et s'exécute toutes les deux périodes. La période de base du séquenceur correspond avantageusement à la période de la tâche dont la période est la plus courte. Cet exemple peut s'étendre naturellement à un nombre quelconque de tâches périodiques dont la période est un multiple de la période de base. Dans le séquençement illustré par la Fig. 1, la tâche T1 est divisée en deux tâches T11 et T12
25 par le concepteur. Lors d'une première période, la tâche T0 est lancée. Lorsqu'elle se termine, le séquenceur effectue un basculement vers la tâche T11. Lorsque celle-ci se termine, il n'y a plus de tâche pour la période courante, le processeur est inactif jusqu'à la période suivante. Au début de la période suivante, la tâche T0 est relancée. Lorsqu'elle se termine, le séquenceur effectue un basculement vers la seconde tâche
30 T12. Ce fonctionnement se répète.

La Fig. 2 illustre le fonctionnement d'un séquenceur préemptif pour les deux mêmes tâches T0 et T1. Dans ce second exemple, le concepteur n'a plus à diviser la tâche T1 en deux tâches T11 et T12. Le séquenceur laisse tourner la tâche T1, nommée par facilité T11 jusqu'à la fin de la première période. Ce n'est qu'à la fin de

la première période que le séquenceur préempte la tâche T1 pour effectuer un basculement vers la tâche T0. Cette interruption nécessite typiquement la sauvegarde du contexte de la tâche T1 pour permettre sa restauration ultérieure. Lors de la seconde période et lorsque la tâche T0 est terminée, le séquenceur effectue le
5 basculement sur la tâche T1, il restaure celle-ci et la relance. Le fonctionnement diffère du précédent en ce que c'est la fin de la période qui détermine la division de la tâche T1 en deux segments temporels T11 et T12. Ces segments vont différer d'une période à une autre du fait de la variabilité du temps d'exécution de T0.

En effet, la durée d'exécution d'une tâche n'est pas fixe. Elle varie en fonction
10 de plusieurs facteurs, le principal étant qu'elle dépend des données effectivement traitées qui varient typiquement d'une instance d'exécution à une autre. Par exemple, une branche du code sera empruntée ou non selon la valeur d'une donnée externe. La durée précise de la tâche T0 est donc variable. De ce fait, le moment de lancement de la tâche T1 est également variable, et le traitement effectué dans T1 à la fin de la
15 période qui déclenche l'interruption de la tâche T1 est donc variable lui aussi.

Pour pouvoir garantir le bon fonctionnement du système, il est obligatoire d'éviter qu'un dysfonctionnement d'une tâche soit perçu comme un dysfonctionnement d'une autre tâche, car cela pourrait déclencher des modes dégradés inappropriés. La Fig. 3 illustre un tel problème. On se place dans le cas d'un
20 séquenceur fonctionnant selon le mode illustré par la Fig. 2. On regarde plus précisément ce qui se passe entre l'instant 2 et l'instant 4. On voit que l'instance d'exécution de la tâche T0 démarrée à l'instant 2 et référencée T0c se prolonge de manière inhabituelle presque jusqu'à la fin de la période. Elle se termine tout de même et le séquenceur effectue le basculement sur la tâche T1 qui est presque aussitôt interrompue à l'instant 3. A la période suivante, on constate l'exécution normale d'une
25 instance de la tâche T0 puis le basculement normal vers la tâche T1 ici référencée T12c. Cette tâche va rencontrer un problème à l'instant 4, car elle n'est pas terminée du fait du temps volé par T0c lors de la période précédente. On voit que ce vol de temps par T0 n'influe pas négativement le fonctionnement de T0, mais provoque un
30 dysfonctionnement de T1 lors de la période suivante. T1 qui dispose de deux périodes pour s'exécuter se trouve interrompue une nouvelle fois et n'a pas le temps de se terminer.

Pour résoudre ce problème, l'état de l'Art se base sur la détermination d'une limite de durée pour l'exécution des traitements d'une tâche appelée « deadline ».

Cette limite de durée est allouée pour chaque instance d'exécution de cette tâche, et si elle est atteinte, c'est le signe d'un dysfonctionnement et le séquenceur lance alors la tâche suivante. Elle est déterminée à l'aide de l'estimation d'un déroulement d'un pire cas. Le mode particulier de détermination est en dehors de la description de ce document.

L'invention ajoute à cet état de l'Art le fait de rendre constante la durée d'exécution des tâches : elles ont alors comme durée leur propre deadline. Le séquenceur est alors programmé pour n'effectuer le basculement entre deux tâches qu'une fois la limite de durée associée à une tâche donnée écoulée et non pas dès la fin d'exécution des traitements de la tâche. On rend ainsi déterministe le moment de basculement entre les tâches.

La Fig. 4 illustre un tel fonctionnement. La limite de durée associée à l'exécution de la tâche T0 est appelée D0. Celle qui est associée à la tâche T1 est appelée D1. La Fig. 4 ne montre que les deux premières périodes de fonctionnement du séquenceur, ce fonctionnement se répète à l'identique ensuite. La tâche T0 est lancée à l'instant 0, D0 indiquant la fin de sa limite de durée. Ses traitements se terminent effectivement un peu avant D0. Le séquenceur n'effectue pas de basculement vers la tâche T1 avant D0. Il introduit une période d'attente L0 pour que la limite de durée allouée à la tâche s'écoule avant de déclencher le basculement. Le fonctionnement du processeur entre la fin des traitements de la tâche T0 et la limite de durée D0 est libre. Par exemple, on peut le mettre en sommeil, effectuer une boucle d'attente ou effectuer des auto-tests jusqu'à la limite de durée D0. Lorsque cette limite est atteinte, le séquenceur provoque le basculement vers la tâche T1. L'exécution de T1 ou plus précisément du premier segment T11 de T1 est interrompue par la fin de la période à l'instant 1. Le séquenceur provoque alors le basculement vers une nouvelle occurrence d'exécution de la tâche T0. Sur la figure, cette occurrence dure moins longtemps que la première. Elle se prolonge par une période d'attente L0 jusqu'à la limite de durée D0 liée à cette nouvelle occurrence. Le séquenceur provoque alors un nouveau basculement vers la tâche T1 précédemment interrompue et lance le second segment T12 de la tâche T1. Celui-ci se prolonge par une période d'attente L1 pour attendre la limite de durée D1 associée à la tâche T1.

On constate que dans ce mode de fonctionnement, un dysfonctionnement de la tâche T0 qui viendrait prolonger de manière anormale la durée de fonctionnement d'une occurrence de la tâche provoquerait une interruption anormale de cette dernière

lorsque la limite de durée D0 associée expire. La tâche T1 est lancée en tout état de cause à l'instant D0 sans être influencée par le dysfonctionnement de T0. Les moments où le séquenceur provoque un basculement entre les tâches sont ici déterministes.

5 Lorsque l'on s'intéresse au fonctionnement d'un tel système lorsque le processeur dispose d'une mémoire cache, on constate que la politique de séquençement précédente, bien qu'assurant des instants de commutation déterministes, ne suffit pas à assurer l'indépendance d'exécution des tâches : l'existence même des caches et le mécanisme de gestion du cache de données vont
10 entraîner dans certains cas une influence du fonctionnement d'une tâche sur la vitesse d'exécution d'une autre tâche. Par exemple, sur la Fig. 4, au début de la seconde exécution de T0, l'état des caches en ce qui concerne les instructions et données utiles à T0 est complètement dépendant des traitements qu'aura effectué le segment T11.

Il existe deux modes de fonctionnement d'une mémoire cache. Un premier
15 mode est appelé *write-through* en anglais, nous l'appellerons mode transparent dans ce document. Dans ce mode, une donnée est écrite en même temps dans la mémoire cache et dans la mémoire principale. Le contenu de la mémoire cache est toujours cohérent avec celui de la mémoire principale. Ce mode transparent de gestion a l'avantage de faciliter le fonctionnement des architectures multi-processeurs, car il
20 résout les problèmes de cohérence, mais il est en général moins rapide que l'autre mode, en particulier en cas d'écritures en rafale, car les données sont systématiquement écrites dans la mémoire principale.

Un second mode de fonctionnement est appelé *write-back* en anglais, nous l'appellerons mode retardé dans ce document. Dans ce mode, les données sont écrites
25 uniquement dans la mémoire cache. A chaque emplacement de la mémoire cache est associé un bit de cohérence (*dirty bit* en anglais) qui est mis à un lors de l'écriture. Ce bit marque que la donnée mémorisée n'est pas cohérente avec la mémoire principale et doit être recopiée en mémoire principale. Cette recopie s'effectue lorsque le processeur a besoin de récupérer un emplacement dans le cache. Cette recopie assure
30 la mise en cohérence de la mémoire cache et de la mémoire principale, le bit de cohérence est alors remis à zéro. Ce mode de fonctionnement est plus rapide du fait que la recopie effective des données du cache en mémoire n'a lieu que lorsqu'elle est vraiment nécessaire. C'est pour cette raison qu'il est appelé mode retardé dans ce document, l'écriture en mémoire étant retardée.

Il existe une opération de gestion de la mémoire cache particulière qui se nomme invalidation du cache. Elle consiste en une réinitialisation complète et instantanée du cache qui revient dans l'état où il était à la mise sous tension, c'est-à-dire qu'il ne contient plus aucune information. Elle permet de vider la mémoire cache de manière rapide lorsque l'on est sûr que les données qu'elle contient sont obsolètes et ne nécessitent pas de recopie en mémoire. En effet, l'invalidation du cache provoque la perte des données qui n'ont pas été recopiées en mémoire.

Avantageusement, le séquenceur introduit une étape d'invalidation de la mémoire cache lors d'une opération de basculement entre deux tâches. Cette opération rend indépendante des autres tâches l'exécution de la tâche, car chaque segment démarre sur un cache vide. Par contre, du fait de cette invalidation, il est nécessaire d'assurer la cohérence entre la mémoire cache et la mémoire principale lorsque le séquenceur reprend le contrôle. Cette mise en cohérence, contrairement à l'invalidation qui est instantanée, prend un temps non négligeable proportionnel au nombre d'emplacements non cohérents (*dirty* en anglais) du cache. Elle est facile à réaliser en fin de traitement pour une tâche non interrompue. Par contre, une tâche interrompible, lors de son exécution, ne sait pas facilement anticiper l'instant d'interruption. En pratique, cela implique un choix entre deux stratégies : soit les tâches interrompibles doivent fonctionner en mode transparent, soit le séquenceur doit, avant l'invalidation, mettre en cohérence le cache lorsqu'il préempte une tâche qui utilise le mode retardé. Ces deux stratégies posent chacune un problème d'efficacité.

En ce qui concerne la première stratégie, le problème est le mode transparent lui-même qui est pénalisant en temps d'exécution. Or, comme les basculements entre tâches ont été rendus déterministes par l'introduction des limites de durée imposées à chaque tâche, il est facile de déterminer lorsque l'on lance une tâche si on lance le dernier segment de cette tâche ou pas : il suffit de constater que la limite de durée de la tâche est dans la période courante ou pas. Si la nouvelle tâche active est amenée à se terminer avant l'échéance de la période de séquençement courante, le séquenceur autorise avantageusement le mode retardé, sinon il force le mode transparent. Si l'on se réfère à la Fig. 4, on constate que lorsqu'on lance la tâche T1 dans la première période entre l'instant 0 et l'instant 1, on lance le segment T11, la limite de durée D1 n'appartient pas à la période courante. On ne lance donc pas le dernier segment de la tâche : le mode transparent est forcé. Par contre, dans la seconde période, entre l'instant 1 et l'instant 2, la limite de durée D1 appartient à la période courante. On sait

donc que lorsqu'on lance le segment T12 de la tâche T1 on lance le dernier segment : le mode retardé est autorisé et sera donc appliqué si la tâche est configurée en mode retardé. Bien sûr, lorsque les traitements de la tâche sont terminés, la tâche doit mettre en cohérence la mémoire cache avec la mémoire principale. A cette condition, on est
5 sûr que cette mémoire cache est propre au prochain basculement. De ce fait, on autorise avantageusement un mode retardé de gestion du cache lorsque l'on lance le dernier segment d'une tâche.

En ce qui concerne la seconde stratégie, le problème est la forte variabilité du temps de mise en cohérence du cache, lorsque cette mise en cohérence est exécutée
10 par le séquenceur. On notera que lorsqu'elle est exécutée en fin de tâche, elle ne pose pas problème car sa variabilité sera intégrée dans le temps d'exécution de la tâche, et donc masquée par la limite de durée. Par contre, dans le séquenceur, cela va entraîner une variabilité significative des instants de commutation ultérieurs, donc la perte du séquençement déterministe. Avantageusement, le séquenceur effectue donc une mise
15 en cohérence à durée constante, donc à durée maximale. Autrement dit, on pourrait la considérer comme une mini-tâche avec sa propre limite de durée.

Cette durée constante de mise en cohérence est contrebalancée par le gain généré par l'exécution du segment précédent en mode retardé, sauf lorsque ce segment est court : dans ce cas, l'effet peut être très néfaste. Or, le séquenceur connaît la durée
20 et le numéro du segment qu'il lance. Avantageusement, il peut donc forcer le mode transparent sur un segment non-final jugé trop court, ou selon un choix prédéfini fonction du numéro, et ainsi, lorsque ultérieurement il le préemptera, il n'aura pas à mettre le cache en cohérence.

On voit que la seconde stratégie est en fait une extension de la première : la
25 première autorise le mode retardé sur les segments finaux, la seconde l'autorise en plus sur des segments non-finaux choisis. La seconde permet une utilisation maximale du mode retardé tout en assurant l'indépendance de l'exécution des tâches.

Le fonctionnement du séquenceur ici décrit pour le séquençement de deux
30 tâches pour des raisons de clarté s'étend naturellement à un nombre quelconque de tâches.

REVENDICATIONS

1/ Procédé de séquençement d'un ensemble de tâches périodiques sur un processeur de traitement de l'information, chaque tâche ayant une période propre, le
5 séquençement étant périodique et préemptif, signifiant que le séquençement provoque le basculement d'une tâche active vers une autre tâche à l'échéance de chaque période de séquençement et à la fin de l'exécution d'une instance d'une tâche, caractérisé en ce qu'il comporte les étapes suivantes :

- pour chaque tâche la détermination d'une limite de durée allouée à l'exécution
10 d'une instance de cette tâche ;
- lorsque l'exécution d'une instance d'une tâche s'achève est introduite une étape d'attente que la limite de durée allouée à ladite tâche s'écoule avant de provoquer le basculement.
- lors d'un basculement, la mémoire cache est invalidée.

15

2/ Procédé de séquençement selon la revendication 1, caractérisé en ce que le processeur possédant une mémoire cache pouvant fonctionner dans un mode transparent et dans un mode retardé, le procédé comporte en outre lors du basculement d'une tâche active vers une nouvelle tâche :

- une étape consistant à forcer le mode transparent de la nouvelle tâche active si
20 ce n'est pas son segment final.

3/ Procédé de séquençement selon la revendication 2, caractérisé en ce qu'il comprend en outre :

- lors de la préemption d'une tâche en mode retardé vers une nouvelle tâche, une
25 étape de mise en cohérence du cache en un temps constant, avant l'étape d'invalidation ; et
- sur des segments non-finaux choisis, l'omission de l'étape consistant à forcer le mode transparent de la nouvelle tâche active.

30

4/ Dispositif de traitement de l'information comportant :

- au moins un processeur capable d'exécuter des tâches ;
- un moyen de séquençement périodique et préemptif signifiant que le
séquençement provoque le basculement d'une tâche active vers une autre tâche à

l'échéance de chaque période de séquençement et à la fin de l'exécution d'une instance d'une tâche, pour le séquençement d'un ensemble de tâches périodiques, chaque tâche ayant une période propre ;

caractérisé en ce qu'il dispose en outre :

- 5 - de moyens de prise en compte pour chaque tâche d'une limite de durée allouée à l'exécution d'une instance de cette tâche ;
 - de moyens d'introduire une étape d'attente que la limite de durée allouée à une tâche s'écoule avant de provoquer le basculement lorsqu'une instance de ladite tâche s'achève ;
- 10 - de moyens d'invalider la mémoire cache lors d'un basculement.

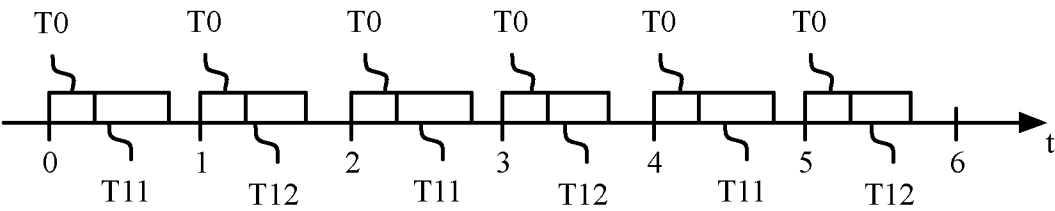


Fig. 1

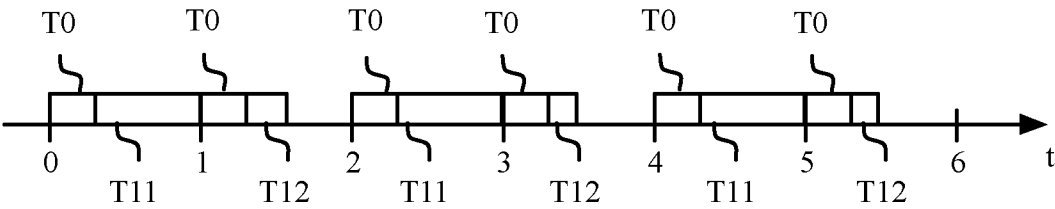


Fig. 2

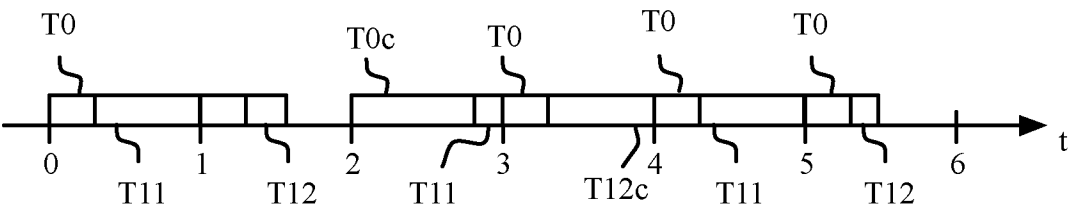


Fig. 3

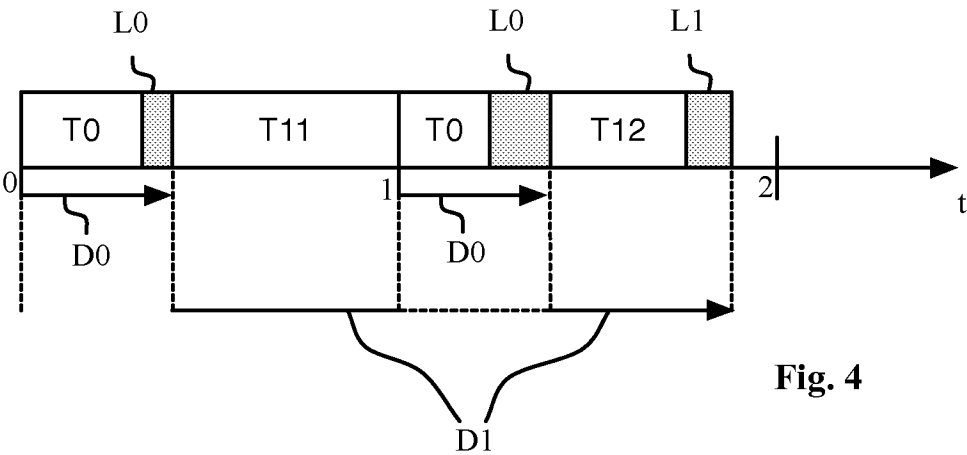


Fig. 4

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2011/052833A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/48 G06F12/08
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 2008/125664 A1 (WAVECOM [FR]; DIDI THIERRY [FR]; MONTES JACQUES [FR]) 23 October 2008 (2008-10-23)	1,2,4
A	* abstract; claim 1 page 12, lines 5-29; figure 1 page 14, line 11 - page 15, line 2; figure 2 page 16, lines 1-12,23 - page 17, line 13; figure 4 ----- -/--	3



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

21 March 2011

Date of mailing of the international search report

30/03/2011

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

de Junca, Irène

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2011/052833

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	A. MARTI CAMPOY, A. PERLES IVARS, J. V. BUSQUETS-MATAIX: "DYNAMIC USE OF LOCKING CACHES IN MULTITASK, PREEMPTIVE REAL-TIME SYSTEMS", 15TH TRIENNIAL WORLD CONGRESS OF THE INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL, BARCELONA 21-26 JULY 2002, 21 July 2002 (2002-07-21), pages 1-6, XP007915961, Retrieved from the Internet: URL: http://www.nt.ntnu.no/users/skoge/prost/proceedings/ifac2002/data/content/02254/2254.pdf [retrieved on 2010-11-22]	1,2,4
A	pages 1-2, paragraphe 1. " INTRODUCTION " page 2, paragraphe 2. " SYSTEM OVERVIEW " -----	3
Y	EP 1 030 243 A1 (TEXAS INSTRUMENTS FRANCE [FR]; TEXAS INSTRUMENTS INC [US]) 23 August 2000 (2000-08-23)	1,2,4
A	paragraphs [0004] - [0005] paragraphs [0016] - [0020]; figure 1 -----	3
A	SEBEK FILIP: "Cache Memories and Real-Time Systems", MRTC TECHNICAL REPORT 01/37, DEPARTMENT OF COMPUTER ENGINEERING, MÄLARDALEN UNIVERSITY, VÄSTERÅS, SWEDEN MRTC TECHNICAL REPORT 01/37, October 2001 (2001-10), pages I-IV,1-42, XP007915957, , DEPARTMENT OF COMPUTER ENGINEERING, MÄLARDALEN UNIVERSITY, VÄSTERÅS, SWEDEN Retrieved from the Internet: URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.5430&rep=rep1&type=pdf [retrieved on 2010-11-22] pages 4-5, paragraphe 1.2.3 " Writing ",figure 1.4 pages 34-36, paragraphe 2.4 " Cache memories in RTS " pages 55-66, paragraphe 3.2 " Extrinsic interference " ----- -/--	1-4

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2011/052833

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	BUSQUETS-MATAIX J V ET AL: "Using harmonic task-sets to increase the schedulable utilization of cache-based preemptive real-time systems", REAL-TIME COMPUTING SYSTEMS AND APPLICATIONS, 1996. PROCEEDINGS., THIR D INTERNATIONAL WORKSHOP ON SEOUL, SOUTH KOREA 30 OCT.-1 NOV. 1996, LOS ALAMITOS, CA, USA,IEEE COMPUTER. SOC, US, 30 October 1996 (1996-10-30), pages 195-202, XP010202973, DOI: DOI:10.1109/RTCSA.1996.554977 ISBN: 978-0-8186-7626-0 the whole document -----	1-4
A	US 4 954 948 A (HIRA GERALD M [US] ET AL) 4 September 1990 (1990-09-04) * abstract; claim 1 column 3, line 51 - column 5, line 31; figure 2A -----	1,4

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2011/052833

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2008125664	A1	23-10-2008	CN 101836189 A 15-09-2010
			EP 2145255 A1 20-01-2010
			FR 2915006 A1 17-10-2008
			US 2010122263 A1 13-05-2010
EP 1030243	A1	23-08-2000	DE 69903707 D1 05-12-2002
			DE 69903707 T2 10-07-2003
			JP 2000242556 A 08-09-2000
			US 6606687 B1 12-08-2003
US 4954948	A	04-09-1990	NONE

PCT/EP2011/052833

C(suite). DOCUMENTS CONSIDERES COMME PERTINENTS		
Catégorie*	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
Y	A. MARTI CAMPOY, A. PERLES IVARS, J. V. BUSQUETS-MATAIX: "DYNAMIC USE OF LOCKING CACHES IN MULTITASK, PREEMPTIVE REAL-TIME SYSTEMS", 15TH TRIENNIAL WORLD CONGRESS OF THE INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL, BARCELONA 21-26 JULY 2002, 21 juillet 2002 (2002-07-21), pages 1-6, XP007915961, Extrait de l'Internet: URL: http://www.nt.ntnu.no/users/skoge/prost/proceedings/ifac2002/data/content/02254/2254.pdf [extrait le 2010-11-22]	1,2,4
A	pages 1-2, paragraphe 1. " INTRODUCTION " page 2, paragraphe 2. " SYSTEM OVERVIEW " -----	3
Y	EP 1 030 243 A1 (TEXAS INSTRUMENTS FRANCE [FR]; TEXAS INSTRUMENTS INC [US]) 23 août 2000 (2000-08-23)	1,2,4
A	alinéas [0004] - [0005] alinéas [0016] - [0020]; figure 1 -----	3
A	SEBEK FILIP: "Cache Memories and Real-Time Systems", MRTC TECHNICAL REPORT 01/37, DEPARTMENT OF COMPUTER ENGINEERING, MÄLARDALEN UNIVERSITY, VÄSTERÅS, SWEDEN MRTC TECHNICAL REPORT 01/37, octobre 2001 (2001-10), pages I-IV,1-42, XP007915957, , DEPARTMENT OF COMPUTER ENGINEERING, MÄLARDALEN UNIVERSITY, VÄSTERÅS, SWEDEN Extrait de l'Internet: URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.5430&rep=rep1&type=pdf [extrait le 2010-11-22] pages 4-5, paragraphe 1.2.3 " Writing ",figure 1.4 pages 34-36, paragraphe 2.4 " Cache memories in RTS " pages 55-66, paragraphe 3.2 " Extrinsic interference " ----- -/--	1-4

C(suite). DOCUMENTS CONSIDERES COMME PERTINENTS		
Catégorie*	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
A	BUSQUETS-MATAIX J V ET AL: "Using harmonic task-sets to increase the schedulable utilization of cache-based preemptive real-time systems", REAL-TIME COMPUTING SYSTEMS AND APPLICATIONS, 1996. PROCEEDINGS., THIR D INTERNATIONAL WORKSHOP ON SEOUL, SOUTH KOREA 30 OCT.-1 NOV. 1996, LOS ALAMITOS, CA, USA, IEEE COMPUTER. SOC, US, 30 octobre 1996 (1996-10-30), pages 195-202, XP010202973, DOI: DOI:10.1109/RTCSA.1996.554977 ISBN: 978-0-8186-7626-0 le document en entier -----	1-4
A	US 4 954 948 A (HIRA GERALD M [US] ET AL) 4 septembre 1990 (1990-09-04) * abrégé; revendication 1 colonne 3, ligne 51 - colonne 5, ligne 31; figure 2A -----	1,4

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Demande internationale n°

PCT/EP2011/052833

Document brevet cité au rapport de recherche		Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
WO 2008125664	A1	23-10-2008	CN 101836189 A	15-09-2010
			EP 2145255 A1	20-01-2010
			FR 2915006 A1	17-10-2008
			US 2010122263 A1	13-05-2010

EP 1030243	A1	23-08-2000	DE 69903707 D1	05-12-2002
			DE 69903707 T2	10-07-2003
			JP 2000242556 A	08-09-2000
			US 6606687 B1	12-08-2003

US 4954948	A	04-09-1990	AUCUN	
