

⑫

DEMANDE DE BREVET D'INVENTION

A1

⑭ Date de dépôt : 09.08.90.

⑮ Priorité :

⑯ Date de la mise à disposition du public de la demande : 14.02.92 Bulletin 92/07.

⑰ Liste des documents cités dans le rapport de recherche : *Se reporter à la fin du présent fascicule.*

⑱ Références à d'autres documents nationaux apparentés :

⑴ Demandeur(s) : BULL (S.A.) — FR.

⑵ Inventeur(s) : Bonnafoux Jean-François et Flon Robert.

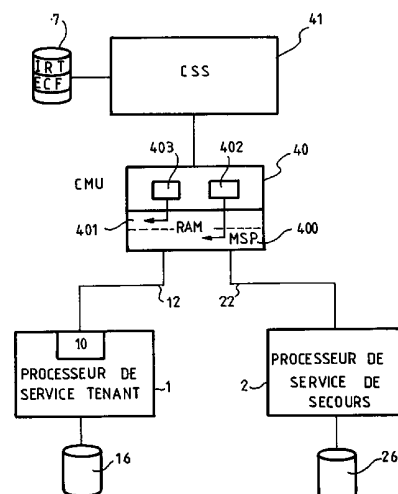
⑶ Titulaire(s) :

⑷ Mandataire : Debay Yves Bull S.A.

⑸ Architecture pour échange dynamique de processeur de service.

⑹ La présente invention concerne une architecture pour échange dynamique de processeur de service.

Architecture pour échange dynamique de processeur de service, comportant un premier processeur de service relié par un réseau et une unité de maintenance (CMU) à un système central (4), ledit processeur de service (1) est relié à un deuxième processeur de service (2) par le réseau et l'unité de maintenance (CMU), caractérisée en ce que chaque processeur de service comporte en plus des programmes de système d'exploitation, un programme superviseur et au moins un service décomposé en deux applications, l'une "corps" composée de l'algorithme du programme, l'autre "présentation" comportant l'interface avec l'opérateur permettant un affichage du type à fenêtres avec une barre de menu, et des moyens MSH (10, 20) de traitement des communications avec les autres processeurs de service, par l'intermédiaire de l'unité de maintenance (CMU), un programme superviseur gérant les numéros d'instance des services et démarrant les services, ledit programme superviseur ayant accès à une table de configuration du système (IRT, ECF).



FR 2 665 778 - A1



ARCHITECTURE POUR ECHANGE DYNAMIQUE DE PROCESSEUR DE SERVICE.

5 La présente invention concerne une architecture pour échange dynamique de processeur de service.

10 Dans une configuration redondante d'un système, il y a deux processeurs de service : l'un est le tenant, c'est celui qui exécute les fonctions de service à un instant donné ; l'autre est celui de secours disponible pour exécuter les fonctions de service au cas où le tenant viendrait à défaillir. Comme les matériels deviennent de plus en plus fiables, le processeur de service de secours n'est pas très souvent sollicité (il reste cependant
15 utile pour une disponibilité du système avoisinant les 100 %). Le processeur de service de secours est donc inutilisé la plupart du temps.

20 Pour améliorer l'opérabilité globale du processeur de service, on veut utiliser le processeur de service de secours, dans ses moments de non activité, non pas en tant que tel, mais en tant qu'écran pour donner la possibilité à un opérateur d'opérer une fonction de service à partir de l'écran du tenant et à un autre
25 opérateur d'opérer une autre fonction de service à partir de l'écran du processeur de service de secours.

30 Cela est possible si l'on dispose d'un environnement multi-tâche, multi-fenêtrage du type WINDOWS, qui permet d'associer à chaque fonction de service une fenêtre.

35 Si de plus, on adopte l'architecture proposée dans le brevet, il suffit de faire tourner la présentation d'un service dans celui de secours pour opérer le service à partir de l'écran et du clavier du processeur de service de secours. Si celui-ci vient à défaillir, la présentation est dynamiquement transférée dans le

processeur de service tenant sans interruption du service.

5 Par ailleurs, si la description d'un système est connue à travers une table, il est nécessaire que le processeur de service connaisse cette table pour procéder aux opérations de service : initialisation du système, reconfiguration du système.

10 On pourrait configurer le disque du processeur de service en copiant la table de configuration du système dans un de ses fichiers. L'inconvénient majeur d'une telle solution réside dans le fait que le processeur de service devient dépendant du système. En particulier, on ne peut
15 pas procéder à un "échange standard" de processeur de service. Il faut procéder à une configuration après échange.

20 Un premier but de l'invention est de proposer une architecture permettant un échange dynamique de processeur de service.

25 Ce but est atteint par le fait que l'architecture, permettant un échange dynamique de processeur de service, comporte un premier processeur de service relié par un réseau et une unité de maintenance du système central (CMU) à un système central (4), ledit processeur de service (1) est relié à un deuxième processeur de service (2) de réserve par le réseau et l'unité de maintenance
30 (CMU) est caractérisée en ce que chaque processeur de service comporte en plus des programmes de système d'exploitation, un programme superviseur et au moins un service décomposé en deux applications, l'une "corps" composée de l'algorithme du programme, l'autre
35 "présentation" comportant l'interface avec l'opérateur permettant un affichage du type à fenêtres avec une barre de menu, et des moyens MSH (10, 20) de traitement des communications avec les autres processeurs de service,

par l'intermédiaire de l'unité de maintenance (CMU), un programme superviseur gérant les numéros d'instance des services et démarrant les services, ledit programme superviseur ayant accès à une table de configuration du système.

Selon une autre caractéristique, l'architecture est caractérisée en ce que l'unité de maintenance comprend une mémoire vive qui communique par l'intermédiaire du système central avec un moyen de stockage du système central, qui contient la table de configuration du système IRT (table d'installation des ressources), et en ce que le superviseur du processeur de service tenant et du processeur de service de réserve contient une routine LBCAM de méthode d'accès à la CMU pour accéder à la table de configuration du système contenue dans le moyen de stockage du système central, la transférer dans son propre disque (16) et la charger dans la mémoire vive de l'unité de maintenance.

Selon une autre caractéristique, l'architecture est caractérisée en ce que le programme superviseur du processeur de service de réserve contient une routine pour, en cas de défaillance signalée par un événement envoyé par la CMU au processeur de service de réserve, lancer le programme superviseur du processeur de service de réserve, et par la routine LBCAM d'effectuer la lecture de la table de configuration IRT dans la mémoire vive de l'unité de maintenance pour la recopier dans le disque dur (26) du processeur de service de réserve.

Selon une autre caractéristique, la mémoire vive est divisée en deux parties ; l'une accessible uniquement par un processeur de l'unité de maintenance CMU, l'autre MSP, accessible uniquement par le coprocesseur de l'unité de maintenance CMU, en exécution de messages envoyés par la routine de méthode d'accès à la CMU lancée par un service tournant sur le processeur de service.

Selon une autre caractéristique, la partie MSP de la mémoire CMU est divisée en trois zones :

- 5 - une première contenant un mot indiquant la longueur
actuelle de la mémoire ;
- une deuxième zone répertoire contenant les étiquettes
des données ;
- 10 - une troisième zone contenant les données.

Selon une autre caractéristique, les étiquettes de la
deuxième zone comprennent le nom du fichier, son espace
15 alloué, sa date de création, sa taille actuelle ;

une information indiquant si l'espace alloué est fixe ou
variable ;

- 20 et un drapeau, dont la valeur 1 indique une session
d'écriture non valablement exécutée.

Selon une autre caractéristique, la routine de la méthode
d'accès affiche le message "fichier non trouvé" dans le
25 cas où le drapeau était à la valeur 1 et procède à la
recopie du fichier, soit de la CMU dans le disque, soit
en sens inverse.

Selon une autre caractéristique, la routine de la méthode
30 d'accès affiche le message "fichier inconnu" dans le cas
où le nom du fichier n'existe pas dans le répertoire de
la CMU et procède à sa recopie s'il existe dans le disque
du processeur de service.

35 Selon une autre caractéristique, les fichiers gérés par
la méthode d'accès de la CMU sont recopiés du disque dur
vers la mémoire vive de la CMU lorsque la plaque
d'interface matériel a été mise hors tension.

Selon une dernière caractéristique, les fichiers gérés par la méthode d'accès de la CMU sont copiés de la mémoire vive de la CMU vers le disque dur d'un processeur de service lorsqu'un processeur de service est initialisé.

D'autres caractéristiques et avantages apparaîtront à la lecture de la description ci-après faite en référence aux dessins annexés, dans lesquels :

- la figure 1 représente un schéma d'ensemble du système ;
- la figure 2 représente les moyens matériels et des logiciels mettant en communication certaines parties du système ;
- la figure 3 représente le principe de l'architecture de communication et de présentation utilisée dans l'invention ;
- la figure 4 représente le schéma d'architecture des logiciels nécessaires à la mise en oeuvre de l'invention ;
- la figure 5 représente le schéma de principe de l'architecture matérielle nécessaire à la mise en oeuvre de l'invention.

30

L'architecture du système est constituée comme représenté à la figure 1 par un processeur de service tenant (1), relié par un logiciel d'interface (10) MSH (Maintenance Station Handler) avec une ligne de traitement de la maintenance des stations de télécommunication (12) formant un réseau de type STARLAN. Cette ligne (12) est reliée à une unité d'horloge et de maintenance CMU (40) qui assure les communications entre le processeur de

35

service (1), un deuxième processeur de service de réserve (2), par l'intermédiaire d'une deuxième ligne (22), et un système central (41). L'unité de maintenance et d'horloge (CMU) communique également par une ligne (43) avec un ensemble de processeurs de maintenance (42). Le processeur de service tenant (1) et le processeur de service de réserve (2) communiquent également par l'intermédiaire d'une interface de gestion de lignes éloigné (11) RLH (Remote Line Handler), et respectivement (21), avec un commutateur de console de service (3), qui permet par l'intermédiaire d'un circuit (30), et d'une matrice de commutation appropriée, de mettre en communication chacun des processeurs de service avec le réseau de télécommunication d'un pays donné, et à travers ce réseau avec une station (5) constituant une console de service éloignée communiquant avec ce réseau (52) par l'intermédiaire d'un logiciel d'interface RSC (50). De même, chacun des processeurs de service peut être mis en communication, à travers le réseau (52), avec une station de maintenance éloignée (6) communiquant avec le réseau (52) par un logiciel d'interface RMS (51). Les lignes (23) et (13) reliées au commutateur de console de service par les logiciels d'interface (21) et (11) ont des débits de 4.800 bauds, tandis que les lignes (12) et (22) formant le réseau "STARLAN" avec l'unité d'horloge et de maintenance (40) ont des débits de l'ordre du mégabit par seconde. Les processeurs de maintenance (42) sont reliés à des ensembles d'unités d'entrée-sortie pour le système central.

30

La figure 2 représente l'environnement logiciel permettant aux processeurs de service (1, 2) de communiquer par le réseau "STARLAN" avec l'unité de maintenance CMU (40). Le processeur de service tenant comporte en plus du logiciel MSH (10) et d'un système d'exploitation comportant le système d'exploitation du disque MS-DOS, un logiciel multitâches à fenêtre tel que WINDOWS, et une partie de logiciel que l'on décrira

35

ultérieurement. Le processeur de service tenant comporte également un logiciel de communication "NETBIOS" (100) communiquant à travers une carte de réseau type "STARLAN" (14) et un élément de hardware (101) constituant une
5 passerelle (HUB), à travers la ligne (12) du réseau STARLAN, vers une deuxième passerelle (400), un circuit de gestion (401), un logiciel d'interface avec le réseau STARLAN "NETBIOS" (402) et un logiciel d'interface (403) avec le processeur de service RSPI. De façon similaire,
10 le processeur de service de réserve (2) comportera également les mêmes éléments, mais avec le chiffre 2 comme référence pour le chiffre des dizaines ou des centaines. Chaque processeur de service comporte en plus de son système d'exploitation (15, 25) un programme corps
15 pour une application donnée (17, 27, fig. 1) et un programme présentation de l'application (18, 28, fig. 1).

Un logiciel superviseur SPV gère les numéros d'instance des services installés et démarre les services. Les
20 processeurs de service tenant et de réserve s'initialisent séparément en parallèle. La différence entre un processeur de service de réserve et un processeur de service tenant est que le processeur de service tenant accepte le lancement de services alors que
25 le processeur de service de réserve attend un événement de la CMU lui disant de devenir tenant. Ainsi, le superviseur SPV du processeur de service tenant (1) envoie, par l'intermédiaire de la CMU un événement au deuxième processeur de service (2) en cas de défaillance
30 du premier, ce qui permet à ce dernier d'instaurer les mêmes applications.

La figure 3 représente le principe d'une architecture nécessaire à la mise en oeuvre de l'invention dans
35 laquelle le programme corps (17) d'une application donnée, tournant par exemple sur le processeur de service tenant, échange des lignes de texte et des messages d'acquiescement ou de réponse avec le programme de

présentation (18) de l'application. Ces échanges de messages et de lignes de texte se font par l'intermédiaire de modules de bibliothèque de gestion des messages LBXMS (171) et LPXMS (183) qui se trouvent liés respectivement au corps de l'application pour LBXMS, et à la présentation de l'application pour LPXMS. Le programme de présentation de l'application échange avec le logiciel WINDOWS (150), les primitives nécessaires à la gestion des tâches et à l'affichage des fenêtres. L'application présentation aura été développée à l'aide du programme WINDOWS (150) qui peut être vu comme un jeu de quatre composants. Un premier composant noyau (Kernel) (151) assurant la gestion des tâches, les allocations de mémoire, la fonction horloge (TIMER) et les liens dynamiques. Un deuxième élément (152) utilisateur (USER) servant à assurer la gestion des fenêtres et la création de celles-ci. Un troisième élément (GDI) (153) permet de réaliser les fonctions graphiques en vue d'effectuer des dessins, un quatrième élément (154) COM qui est un driver WINDOWS, permet de gérer des lignes asynchrones du type RS 232 et un protocole d'échange de données entre applications DDE (échange dynamique de données) permet la communication entre applications.

La communication avec des lignes de texte permet de réduire la densité des informations par rapport à une communication constituée par un jeu de primitives WINDOWS, qui sont utilisées pour générer un écran, et par conséquent, accélère les échanges.

La figure 4 représente les logiciels nécessaires à la mise en place de moyens permettant l'échange dynamique de processeur de service. La CMU (40) est connectée par une connexion de type STARLAN au logiciel MSH (10) ou (20) selon qu'il appartient au processeur de service tenant ou de réserve. L'application MSH dialogue avec une librairie (ou un utilitaire) VSH (VMP Object Sequence Handler) de traitement de séquence objet du programme VMP constituant

le panneau de maintenance virtuel (Virtual Maintenance Panel). Le service SYC appelle CAM (méthode d'accès au fichier CMU) pour mettre à jour les fichiers IRT. CAM lui-même appelle VSH pour exécuter des VOS (séquences d'objets virtuels) pour aller sauver les fichiers dans la mémoire (400, 401) de la CMU et le disque dur du processeur de service tenant (16) ou de réserve (26). Cet utilitaire comporte deux modules ; l'un LBCAM qui est le "corps", l'autre LPCAM qui le "module de présentation" correspondant et dialoguant avec le logiciel WINDOWS User (152) pour assurer la présentation à l'écran. Le corps du service (105) peut être le corps du programme du panneau de maintenance virtuel (VMP) et la présentation du service (104) peut être la présentation VMP. Le panneau de maintenance (VMP) permet d'utiliser des commandes spécifiques, de visualiser dynamiquement des ressources sélectionnées, construire et compiler des directives et de les exécuter, d'afficher la liste des fichiers, d'utiliser un tuteur pour construire champ par champ une ligne de commandes et permettre des opérations telles que lecture-écriture de registres, lecture-écriture de mémoires, etc.

LBCAM (103) permet d'accéder, grâce au fichier du système MS-DOS (15), au disque dur (16), soit en lecture, soit en écriture selon le cas.

L'architecture nécessaire au processus d'échange dynamique de processeur de service est représentée à la figure 5 dans laquelle le disque (7), disque du système, où se trouvent entre autre les fichiers de configuration (IRT, ECF, RCFT, ROLST, R1LST, EX.LST et DG.LST) et les microprogrammes fonctionnels du processeur du CSS. Au cours de l'initialisation du système, les fichiers du disque (7) sont chargés dans la mémoire du CSS (41) et le programme d'initialisation va charger les deux fichiers de configuration IRT et ECF dans la mémoire du CSS (41), dans le disque dur (16) de la AUSP(1) d'une part, et

d'autre part dans la mémoire (400) de la CMU (40). La mémoire de la CMU comporte deux parties, une (401), l'autre (400) appelée MSP, chacune d'une taille de un mégaoctet. La partie (401) est uniquement accessible par le processeur (403) de la CMU, tandis que la partie MSP (400) est accessible par le coprocesseur (402) ADMA. Ce coprocesseur (402) a un champ d'adresses qui lui permet d'accéder aux adresses d'un méga à deux mégaoctets, alors que le processeur (403) a un champ d'adresses permettant d'accéder aux adresses de zéro à un mégaoctet. L'espace MSP est spécialement réservé à l'usage du processeur de service (1) et permet entre autre de stocker les fichiers IRT et ECF. L'espace mémoire MSP est géré et organisé par le processeur de service (1) et subdivisé en trois zones principales ;

- un premier mot contenant une valeur donnant la longueur actuelle de la mémoire MSP. Ce mot est mis à jour par la CMU et a un attribut d'accès en lecture seulement pour le processeur de service (1) ;

- une seconde zone est constituée par le répertoire contenant les étiquettes des fichiers, ce répertoire étant créé et mis à jour par le corps de l'application méthode d'accès à la CMU (LBCAM) ;

- un champ de données contenant les données des fichiers.

Cette architecture permet d'utiliser le processeur de service de réserve (2) en tant que station permettant de faire tourner la présentation d'une application, tout en autorisant une reprise en main du système par le superviseur du processeur de service de réserve (2), au cas où le processeur de service tenant viendrait à défaillir.

Dans un tel cas, le superviseur SPV (1052) du processeur de service tenant (1) demande le lancement sur le

processeur de service de réserve (2) de l'application corps en cours d'exécution sur le processeur de service tenant (1) avec ou sans interruption de la présentation en cours, selon le cas.

5

Pour ce faire, la table de configuration du système, qui dans l'état de l'art était stockée sur le disque du processeur de service tenant, est, pour les besoins de l'invention, à l'initialisation du système généré sur un disque (7) du système et non sur un disque du processeur de service, puis la table de configuration est transférée vers le processeur de service (1) lors de la première initialisation du système.

10

Le processeur de service (1) copie cette table dans la partie (MSP) de la RAM (400) de la plaque d'interface matériel (40) qui devient la référence, c'est-à-dire que le processeur de service (1) accède la RAM lorsqu'il utilise la table de configuration. Les deux processeurs de service ont chacun un chemin d'accès vers cette RAM.

20

Enfin, cette table est copiée en parallèle sur un fichier du disque (10) processeur de service tenant (1) pour pouvoir éventuellement réinitialiser la RAM (400) lorsque la plaque d'interface matériel (40) a été, pour une raison quelconque, mise hors tension.

25

Cette architecture permet de procéder à un échange dynamique de processeur de service. A la connexion d'un nouveau processeur de service, celui-ci copie la table de configuration de la RAM vers son disque. La même opération est effectuée lorsqu'on bascule du processeur de service tenant (1) au processeur de service de réserve (2).

30
35

Chaque fois qu'un accès à un fichier est demandé, le logiciel CAM accède au fichier d'étiquettes du répertoire de la CMU pour déterminer si un fichier est contenu ou

non. Le format d'étiquette de fichier est constitué par les bits 0 à 7 pour le nom de fichier, par les bits 8 à 10 pour le nom d'un fichier d'extension, par le bit 11 pour désigner si l'espace alloué est fixe ou variable, par les bits 12 à 15 pour indiquer l'espace alloué en nombre d'octets, par les bits 16 à 19 pour indiquer la taille actuelle du fichier en nombre d'octets, par les bits 20 à 23 pour indiquer l'adresse de commencement du fichier, par les bits 24 à 29 pour indiquer la date de création. Enfin, les bits 30 et 31 constituent un drapeau initialisé à zéro et mis à la valeur -1 chaque fois que le fichier est en cours d'écriture. Une fois la session d'écriture terminée, le drapeau est remis à zéro. Ceci constitue une protection contre les fins d'écriture anormales. Si le drapeau n'est pas à zéro quand l'application demande à CAM de lire un fichier, cela signifie que la session d'écriture précédente ne s'est pas exécutée correctement et que le fichier est invalide. Dans ce cas, le module CAM renvoie le code "fichier non trouvé". Dans le cas où le nom d'un fichier n'est pas trouvé par CAM lorsqu'on lui demande d'accéder à un certain fichier, ce dernier renvoie le code "fichier inconnu". Après lecture du répertoire, la méthode d'accès CAM renvoie le message suivant "fichier non trouvé" dans le cas où le fichier n'est présent, ni dans la mémoire de la CMU, ni dans le disque dur du processeur de service tenant. Dans le cas où le fichier est présent en mémoire de la CMU, mais pas dans le disque dur du processeur de service, la méthode d'accès renvoie le message "mémoire CMU.mise à jour du disque dur". Dans le cas inverse, la méthode d'accès renvoie le message "disque dur-processeur de service.mise à jour de la CMU".

Dans le cas où les fichiers sont présents dans la mémoire CMU et dans le disque dur, la méthode d'accès renvoie le message "mémoire CMU". Chaque fois que cela est nécessaire, la méthode d'accès procède à la mise à jour de l'élément correspondant. Outre les fichiers ECF et IRT

- pour les processeurs de service de réserve et tenant, la méthode d'accès gère également la table de reconfiguration du système, les listes de réservation ROLST, R1LST pour les processeurs de service tenant et de réserve, la liste d'exclusion EX.LST et la liste de diagnostic DG.LST, ainsi que les paramètres d'initialisation des processeurs de service S0 et S1 contenus dans le fichier INIT PAR.
- 5
- 10 D'autres modifications à la portée de l'homme de métier font également parties de l'esprit de l'invention.

REVENDEICATIONS

1. Architecture pour échange dynamique de processeur de service, comportant un premier processeur de service
5 relié par un réseau et une unité de maintenance (CMU) à un système central (4), ledit processeur de service (1) est relié à un deuxième processeur de service (2) par le réseau et l'unité de maintenance (CMU), caractérisée en ce que chaque processeur de service comporte en plus des
10 programmes de système d'exploitation, un programme superviseur et au moins un service décomposé en deux applications, l'une "corps" composée de l'algorithme du programme, l'autre "présentation" comportant l'interface avec l'opérateur permettant un affichage du type à
15 fenêtres avec une barre de menu, et des moyens MSH (10, 20) de traitement des communications avec les autres processeurs de service, par l'intermédiaire de l'unité de maintenance (CMU), un programme superviseur gérant les numéros d'instance des services et démarrant les
20 services, ledit programme superviseur ayant accès à une table de configuration du système (IRT, ECF).

2. Architecture selon la revendication 1, caractérisée en ce que l'unité de maintenance (CMU) comprend une mémoire
25 vive (400) qui communique par l'intermédiaire du système central avec un moyen de stockage (7) du système central qui contient la table de configuration du système (IRT), et en ce que le superviseur du processeur de service tenant et du processeur de service de réserve contient
30 une routine (LBCAM) de méthode d'accès à la CMU pour accéder à la table de configuration du système contenue dans le moyen de stockage du système central, la transférer dans son propre disque (16) et la charger dans la mémoire vive (400) de l'unité de maintenance (CMU).

35

3. Architecture selon la revendication 2, caractérisée en ce que le programme superviseur du processeur de service de réserve (2) contient une routine pour, en cas de

défaillance du processeur de service tenant (1) signalée par un événement envoyé par la CMU au processeur de service de réserve (2), lancer le programme superviseur du processeur de service de réserve (2), et par la
5 routine LBCAM effectuer la lecture de la table de configuration IRT dans la mémoire vive (400) de l'unité de maintenance (CMU) pour la recopier dans le disque dur (26) du processeur de service de réserve (2).

10 4. Architecture selon la revendication 3, caractérisée en ce que la mémoire vive est divisée en deux parties, l'une accessible uniquement par un processeur (403) de l'unité de maintenance (CMU), l'autre (MSP) accessible uniquement par un coprocesseur (402) de l'unité de maintenance (CMU)
15 en exécution de messages envoyés par la routine (CAM) de méthode d'accès de la CMU lancée par un service tournant sur le processeur de service (1).

20 5. Architecture selon la revendication 4, caractérisée en ce que la partie MSP (400) de la mémoire CMU est divisée en trois zones :

- une première contenant un mot indiquant la longueur actuelle de la mémoire ;

25

- une deuxième zone répertoire contenant les étiquettes des dossiers ;

- une troisième contenant les données.

30

6. Architecture selon la revendication 5, caractérisée en ce que les étiquettes de la deuxième zone comprennent le nom du fichier, son espace alloué, sa date de création, sa taille actuelle,

35

une information indiquant si l'espace alloué est fixe ou variable,

et un drapeau, dont la valeur 1 indique une session d'écriture non valablement exécutée.

5 7. Architecture selon la revendication 6, caractérisée en ce que la routine de la méthode d'accès CAM affiche le message "fichier non trouvé" dans le cas où le drapeau était à la valeur 1 et procède à la recopie du fichier soit de la CMU dans le disque, soit en sens inverse.

10 8. Architecture selon la revendication 6, caractérisée en ce que la routine de la méthode d'accès CAM affiche le message "fichier inconnu" dans le cas où le nom du fichier n'existe pas dans le répertoire de la CMU et procède à sa recopie du disque dur (16) vers la mémoire
15 (400) de la CMU s'il existe dans le disque dur (16) du processeur de service (1).

20 9. Architecture selon la revendication 8, caractérisée en ce que les fichiers gérés par la méthode d'accès de la CMU sont recopiés du disque dur (16) vers la mémoire vive (400) de la CMU lorsque la plaque d'interface matériel a été mise hors tension.

25 10. Architecture selon la revendication 8, caractérisée en ce que les fichiers gérés par la méthode d'accès de la CMU sont recopiés de la mémoire vive (400) de la CMU vers le disque dur (16, 26) d'un processeur de service (1, 2), lorsque le processeur de service est initialisé.

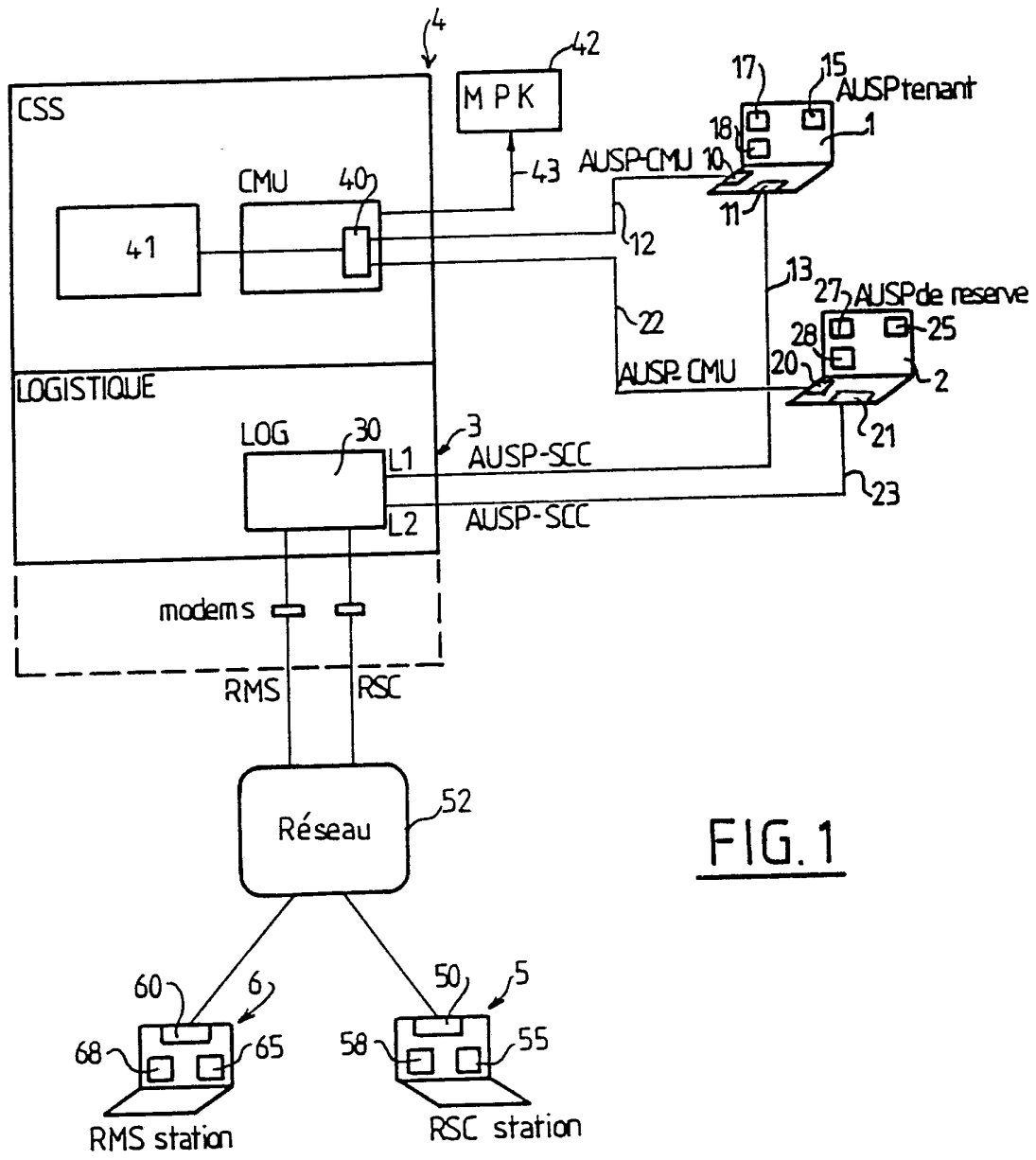


FIG. 1

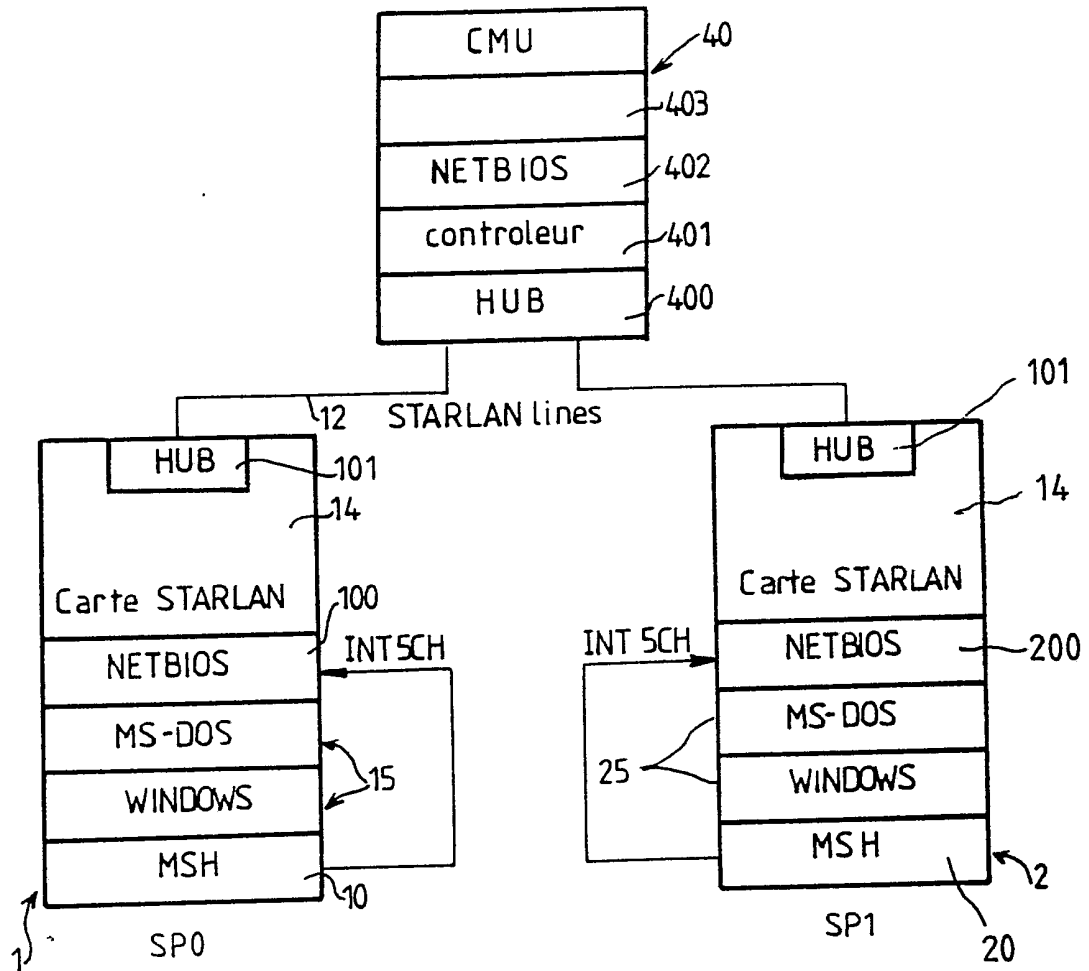


FIG. 2

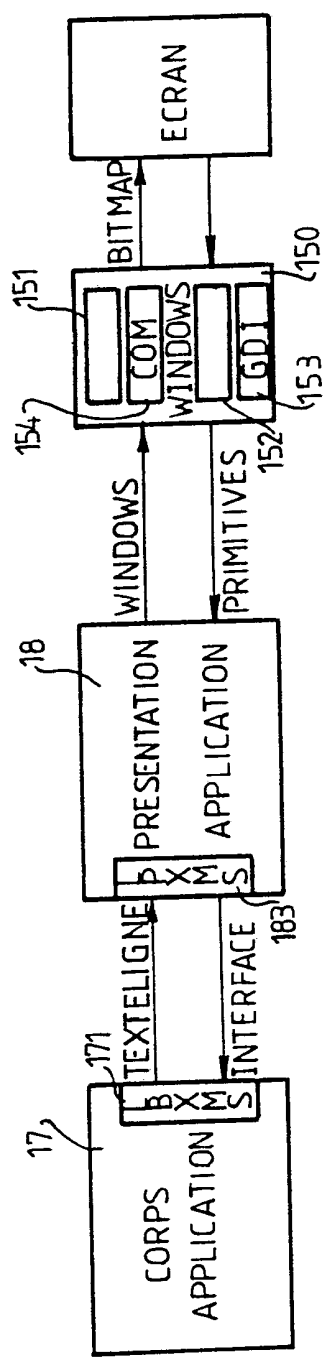


FIG. 3

FIG. 4

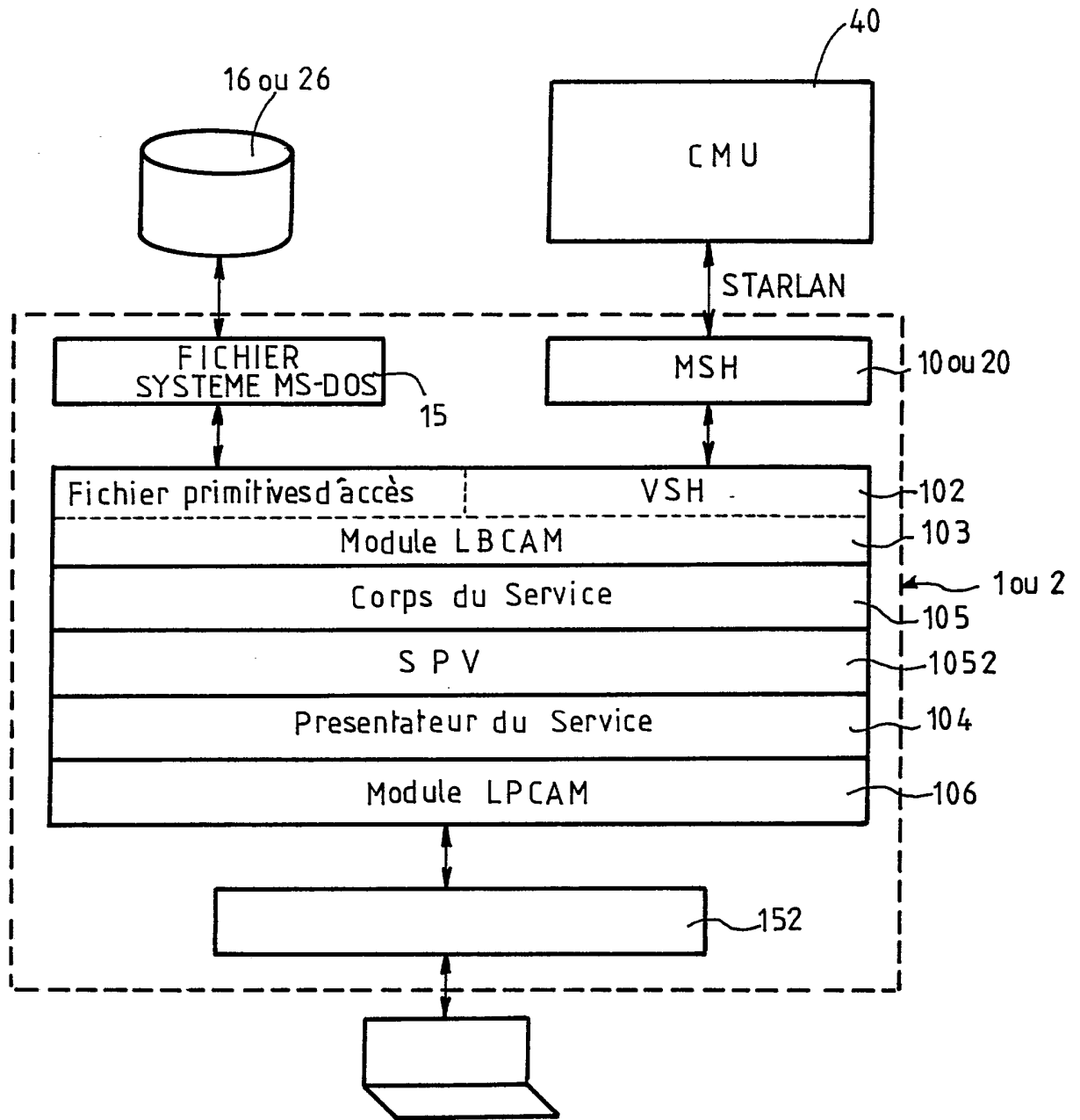
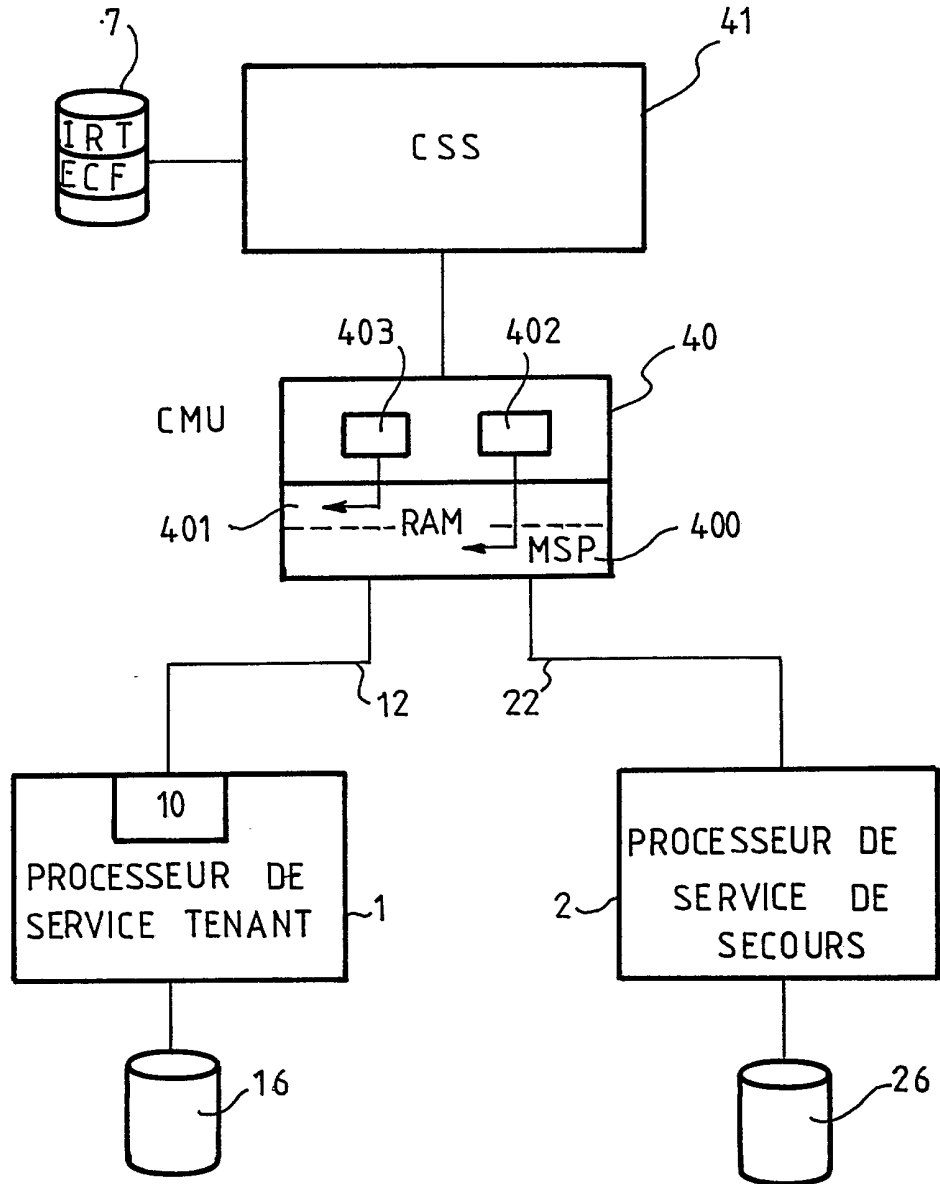


FIG. 5



INSTITUT NATIONAL
de la
PROPRIETE INDUSTRIELLE

RAPPORT DE RECHERCHE
établi sur la base des dernières revendications
déposées avant le commencement de la recherche

N° d'enregistrement
national

FR 9010180
FA 449850

| DOCUMENTS CONSIDERES COMME PERTINENTS | | Revendications concernées de la demande examinée |
|--|--|--|
| Catégorie | Citation du document avec indication, en cas de besoin, des parties pertinentes | |
| A | US-A-4894828 (NOVY ET AL.) * le document en entier * --- | 1 |
| A | EP-A-0083422 (IBM CORP.) * le document en entier * ---- | 1 |
| A | EP-A-0216353 (HITACHI, LTD.) * le document en entier * --- | 1 |
| A | US-A-4014005 (FOX ET AL.) * le document en entier * --- | 1 |
| A | EP-A-0348293 (BULL S.A.) * le document en entier * --- | 1 |
| A | IBM TECHNICAL DISCLOSURE BULLETIN. vol. 29, no. 4, septembre 1986, NEW YORK US page 1614 "ASYMMETRIC MULTIPROCESSING SYSTEM" * le document en entier * --- | 1 |
| A | IBM TECHNICAL DISCLOSURE BULLETIN. vol. 29, no. 4, septembre 1986, NEW YORK US page 1631 "DUAL SERVICE PROCESSOR BACKUP SYSTEM" * le document en entier * ----- | 1 |
| | | DOMAINES TECHNIQUES RECHERCHES (Int. Cl.5) |
| | | G06F |
| Date d'achèvement de la recherche | | Examineur |
| 03 AVRIL 1991 | | DURAND, J |
| <p>CATEGORIE DES DOCUMENTS CITES</p> <p>X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : pertinent à l'encontre d'au moins une revendication ou arrière-plan technologique général O : divulgation non-écrite P : document intercalaire</p> <p>T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure. D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant</p> | | |