8 - MÉMOIRE VIRTUELLE : EFFICACITE

1. CACHE ET MÉMOIRE VIRTUELLE

Avec une mémoire paginée, l'accès à une donnée nécessite deux accès à la mémoire : un pour lire la table des pages, l'autre pour lire la donnée. Pour ne pas diviser par deux les performances du système, on utilise un cache spécial, appelé TLB (Translation Look-aside Buffers), qui permet de stocker le couple <n° de page, case mémoire>. Si le couple correspondant à la page accédée est présent dans la TLB, alors l'accès à la table des pages devient inutile. Typiquement, la TLB contient entre 8 et 2048 entrées.

1.1.

On suppose qu'un accès à la TLB demande 20 ns et qu'un accès mémoire demande 100 ns. Pour une TLB avec un taux de hit de 90%, quel est le temps moyen d'accès à une donnée ?

En plus de la TLB qui permet une translation rapide de l'adresse logique à l'adresse physique, le système dispose d'un cache pour améliorer la vitesse d'accès aux données. Le cache est une mémoire à accès rapide, structurée en lignes. Chaque ligne stocke un bloc de données d'une part, tout ou partie de l'adresse du bloc permettant d'identifier le bloc de façon unique, ainsi que diverses informations relatives à la validité, l'ancienneté, ... dudit bloc.

La combinaison de la mémoire cache et de la mémoire virtuelle est délicate : ou bien le cache contient les adresses physiques des blocs, et dans ces conditions la traduction adresse virtuelle - adresse physique doit être faite entre le processeur et le cache; ou bien le cache contient les adresses virtuelles des blocs, et la traduction adresse virtuelle-adresse physique a lieu entre le cache et la mémoire.

1.2.

Représentez schématiquement l'organisation de la hiérarchie mémoire dans les deux cas. Quels sont les avantages et inconvénients de chaque organisation ?

2. PARTAGE D'INFORMATION EN MÉMOIRE

Le partage d'information entre plusieurs processus soulève deux principaux problèmes :

- le partage physique : comment faire en sorte que les données partagées existent en un exemplaire unique ?
- la protection : comment garantir la cohérence des accès aux informations partagées ?

2.1.

Représentez la topologie des tables de pages de deux processus, A et B, partageant une donnée de la case 17 située aux adresses virtuelles <1,155> pour A et <3,155> pour B.

Soient 2 processus A et B exécutant le même programme. Le segment 0 correspond au code, le segment 1 à la pile et le segment 2 aux données. Seules les pages suivantes ont été chargées (on représente le doublet <n° segment, n° page>) :

pour le processus A : les pages <0,0> dans la case 10, <1,3> dans la case 20 et <2,1> dans la case 30.

pour le processus B : les pages <0,1> dans la case 40 et <1,4> dans la case 50.

2.2.

Quels segments peuvent être partagés entre ces 2 processus ?

2.3.

Faites un schéma des structures de données (tables des segments et tables des pages) pour ces deux processus.

2.4.

Quels nouveaux problèmes sont introduits par le partage d'une case ?

2.5.

Quel est l'influence du partage sur les politiques de remplacement de pages ?

3. GESTION OPTIMISÉ D'UN FORK

Dans UNIX, l'appel système "fork" permet de créer un processus (le fils) à l'image de l'appelant (le père). C'est-à-dire que les segments de code, de pile et de données du processus père sont **copiés** dans l'espace d'adressage du processus fils. Chaque processus (père et fils) évolue ensuite de façon indépendante.

3.1.

Comment peut-on utiliser les mécanismes de partage pour réduire le coût du fork (le coût de la copie) ?

3.2.

Quelles sont les vérifications nécessaires avant l'accès à une page ? Proposez une solution simple pour gérer les conflits d'accès lorsqu'une page est partagée en lecture/écriture par plusieurs processus.

3.3.

On souhaite gérer la mémoire virtuelle pour pouvoir partager les pages tant qu'elles ne sont pas modifiées. Quelles structures de données faut-il ajouter pour savoir si une page doit être recopiée ?

Une faute de protection signifie que le processus a accédé à une page en mémoire (valide) mais que les bits associés à la page ont permis de détecter un problème à traiter. La partie matérielle fournit à la procédure l'adresse virtuelle où la faute s'est produite.

3.4.

Indiquez brièvement l'algorithme de la procédure de traitement des fautes de protection.

```
}
```