

TD 6 – La mémoire virtuelle

Présentation

La mémoire virtuelle permet d'exécuter des programmes dont la taille excède la taille de la mémoire réelle. Pour ceci, on découpe (on « paginée ») les processus ainsi que la mémoire réelle en pages de quelques kilo-octets (1, 2 ou 4 ko généralement).

L'encombrement total du processus constitue l'espace d'adressage ou la mémoire virtuelle. Cette mémoire virtuelle réside sur le disque. On ne charge qu'un sous-ensemble de pages en mémoire. Ce sous-ensemble est appelé l'espace physique (réel).

Lorsqu'une adresse est générée, elle est transcodée, grâce à une table, pour lui faire correspondre son équivalent en mémoire physique. Ce transcodage est effectué par des circuits matériels de gestion : **Memory Management Unit (MMU)**. Si cette adresse correspond à une adresse en mémoire physique, le MMU transmet sur le bus l'adresse réelle, sinon il se produit un défaut de page. Pour pouvoir accéder à la page dont on a généré l'adresse, on devra préalablement la charger en mémoire réelle. Pour cela, on choisit parmi les pages réelles une page « victime »; si cette dernière a été modifiée, on la reporte en mémoire virtuelle (sur le disque) et on charge à sa place la page à laquelle on désirait accéder.

Algorithmes de remplacement de pages

L'algorithme de remplacement de page optimal consiste à choisir comme victime, la page qui sera appelée le plus tard possible. On ne peut malheureusement pas implanter cet algorithme, mais on essaye de l'approximer le mieux possible, avec de résultats qui ont une différence de moins de 1 % avec l'algorithme optimal.

La technique **FirstIn-FirstOut** est assez facile à implanter. Elle consiste à choisir comme victime, la page la plus anciennement chargée.

L'algorithme de remplacement de la page la moins récemment utilisée (**Least Recently Used**) est l'un des plus efficaces. Il nécessite des dispositifs matériels particuliers pour le mettre en œuvre. On doit notamment ajouter au tableau une colonne de compteurs. Par logiciel, on peut mettre en œuvre des versions dégradées.

Autres considérations

L'allocation locale ou globale

Lorsqu'on retire une page de la mémoire centrale, on peut choisir la plus ancienne :

- Du point de vue global (la plus ancienne du système) ;
- Du point de vue local (la plus ancienne du processus) ;

En général, l'allocation globale produit de meilleurs résultats.

La prépagination

Lors du lancement d'un processus ou lors de sa reprise après une suspension, on provoque obligatoirement un certain nombre de défauts de pages. On peut essayer de les limiter en enregistrant, par exemple, l'ensemble de travail avant une suspension.

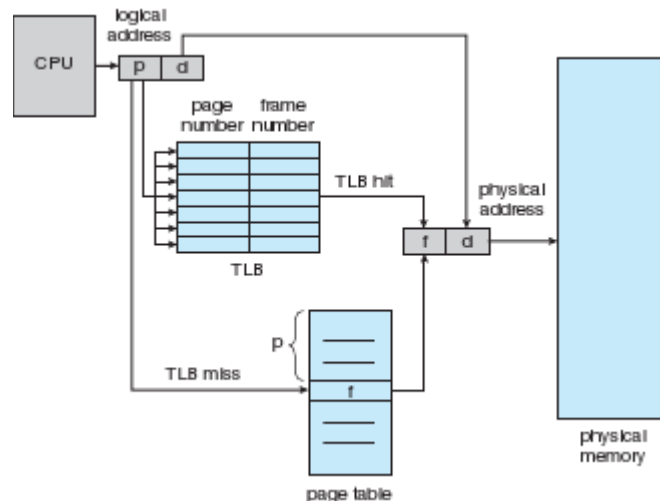
On peut aussi essayer de le deviner. Par exemple, au lancement d'un programme, les premières pages de codes seront vraisemblablement exécutées.

1. Simulateur de gestionnaire de la mémoire virtuelle

Écrivez un programme Java qui simule le transcodage des adresses virtuelles en adresses physiques. Le programme lit le fichier `InputFile.txt` qui contient les adresses virtuelles. Le programme utilise une table de transcodage des adresses virtuelles en adresses physiques – **TLB** et affiche les adresses physiques et le contenu de l'adresse. L'espace d'adressage physique est de 65536 bytes ($=2^{16}$). Le programme lit le fichier contenant plusieurs entières sur 32 bits représentant l'adresse virtuelle. Vous pouvez travailler seulement avec des adresses sur 16 bits, il faut donc tenir compte des 16 LSB. Les 16 bits de l'adresse virtuelle sont divisées en : 8 bits le numéro de page et 8 bits le décalage.

- La dimension de la page : 256 bytes ($=2^8$)
- Le nombre d'entrées dans le table des pages : 256 ($=2^8$)
- Le nombre d'entrées dans TLB est 16 ($=2^4$)
- Le nombre des cadres en mémoire physique est 256 (dans notre cas la taille de la mémoire physique est $256 \times 256 = 64\text{Kb}$)

La translation des adresses



Le fichier d'entrée `InputFile.txt` contient des entières dans l'intervalle 0 – 65366 de la mémoire vituelle. Le programme lit chaque adresse virtuelle et fait le transcodage en adresse physique. Après, le programme lit et affiche le comptenu de l'adresse physique: un byte signé.

L'exécution du programme:

```
java VMsim InputFile.txt
```

ou

```
java VMsim InputFile.txt >> OutputFile.txt
```

Le fichier `OutputFile.txt` contient:

- l'adresse vituelle (celle du fichier `InputFile.txt`)
- l'adresse physique correpondante
- la valeur de byte stoqué à l'adresse physique

Les statistiques:

Quand le programme trouve une adresse virtuelle invalide (le bit correspondant dans la table des pages n'est pas marqué) il s'agit d'un cas de défaut de page. Le progamme doit compter chaque fois quand une entrée dans TLB est trouvee et calculer le pourcentage des pages trouvées vs. pages non trouvées dans TLB.