Système d'Exploitation
Travaux Dirigés (3), Licence 2 Informatique
Gestion Mémoire
Remplacement de pages

L'exécution d'un programme engendre une longue suite d'accès mémoire : il y a souvent plusieurs accès pour une seule instruction et plusieurs dizaines de milliers d'instructions exécutées. L'efficacité d'une politique de remplacement ne peut donc se juger que sur des suites de taille significative, ce qui n'est pas le cas ici. Les exercices proposés dans ce TD ont pour but de vous aider à assimiler les mécanismes des différentes stratégies de remplacement.

Pourquoi l'exécution d'un programme engendre-t-elle (souvent) des défauts de page ? Durant un délai faible, un programme n'utilise qu'une faible partie de son espace de travail. Il est donc inutile et coûteux de charger la totalité de cet espace. Mais en n'en chargeant qu'une partie, il va arriver un moment où l'on accèdera à un élément non chargé. On peut charger cet élément *à la place* d'un élément déjà en mémoire : la stratégie de remplacement va alors déterminer l'élément victime. Dans ce cas, l'espace alloué à la tâche est de taille fixe. On peut aussi *ajouter* cet élément à l'espace de travail. La taille de celui-ci augmente donc, mais comme on ne peut indéfiniment augmenter l'espace de toutes les tâches, il faut à un moment supprimer certains éléments en mémoire. La taille de l'espace varie en fonction des besoins de la tâche, c'est le fonctionnement des stratégies de type *Working Set*.

1. Algorithme optimal

Lors d'un défaut de page, on remplace la page qui sera utilisée le plus tard possible. Il est clair que cet algorithme n'est pas réalisable car il faudrait pour cela connaître l'avenir; cependant, il est d'une grande utilité pour étalonner les autres algorithmes.

On suppose qu'un processus fait référence à ses pages dans l'ordre suivant:

5 0 1 2 0 3 0 4 2 3 0 3 2 1 2

1.1.

> Si on ne limite pas le nombre de cases allouées au processus, combien de défauts de page sont provoqués par cette suite d'accès ? Combien de cases au minimum faut-il allouer au processus pour atteindre ce nombre ?

Mohamed Taghelit , taghelit@univ-tours.fr

1.2.

Donnez l'évolution de la table des pages ainsi que le nombre de défauts de pages si on alloue au processus 3 cases.

2. Algorithme FIFO

Lors d'un défaut de page, on remplace la plus ancienne page qui ait été chargée.

2.1.

On suppose qu'un processus fait référence à ses pages dans l'ordre suivant:

5 0 1 2 0 3 0 4 2 3 0 3 2 1 2

Donnez l'évolution de la table des pages ainsi que le nombre de défauts de pages si on alloue au processus 3 cases.

3. Algorithme FINUFO (first in not used first out)

On suppose qu'il y a un bit R dans le descriptif de chaque page. A chaque accès à une page, le matériel met le bit R à 1. Lorsqu'une page est chargée, le bit R est aussi mis à 1.

Une liste des pages triées selon leur date de chargement est gérée : la page en tête de liste est la plus récemment chargée, celle en fin de liste est la plus anciennement chargée. Lors d'un défaut de page, la page la plus anciennement chargée est examinée. Si son bit R vaut 1, elle est remise en tête de liste et son bit R passe à 0 (elle dispose ainsi d'une deuxième chance). La page déchargée est donc celle qui est la plus anciennement chargée et dont le bit R vaut 0.

3.1.

On suppose qu'un processus fait référence à ses pages dans l'ordre suivant:

5 0 1 2 0 3 0 4 2 3 0 3 2 1 2

Donnez l'évolution de la table des pages ainsi que le nombre de défauts de pages si on alloue au processus 3 cases.

4. Algorithme LRU (Least Recently Used)

Lors d'un défaut de page, on remplace la page la moins récemment utilisée.

4.1.

On suppose qu'un processus fait référence à ses pages dans l'ordre suivant:

5 0 1 2 0 3 0 4 2 3 0 3 2 1 2

Dans le cas où on alloue au processus 3 cases, donnez l'évolution de la liste des pages chargées classées par ordre de dernière utilisation. Quel est le nombre de défauts de pages ?

5. Algorithme par gestion de fenêtre (WS : working set)

Dans cet algorithme une fenêtre (i.e. l'espace de travail ou *working set*) est un intervalle de la suite des accès aux pages. Le nombre de pages contenues dans la fenêtre varie en fonction des accès. Périodiquement, la fenêtre est mise à jour en ne conservant que les pages utilisées durant la période.

5.1.

On suppose qu'un processus fait référence à ses pages dans l'ordre suivant:

5 0 1 2 0 3 0 4 2 3 0 3 2 1 2

En supposant que la fenêtre est mise à jour tous les 3 accès mémoire (on ne conserve donc dans la fenêtre que les pages accédées lors des 3 dernières références), donnez l'évolution de la liste des pages chargées. Quel est le nombre de défauts de pages ?