

GIF-1001 Ordinateurs: Structure et Applications
Solutions : La gestion de la mémoire

1. Décrivez, pour l'allocation contiguë de mémoire avec partitions à taille variables pour placer des processus en mémoire, comment le MMU fait la translation entre l'adresse logique d'une instruction d'un programme son adresse physique en mémoire.

Solution: Le MMU contient une table qui dit où commence chaque processus en mémoire. Pour faire la translation entre une adresse de programme et une adresse de mémoire, il suffit d'additionner l'adresse du début du processus à l'adresse du programme.

2. Supposons une mémoire de 2^N Ko avec des pages de 2^M Ko et des programmes ayant une taille max de 2^O Ko. Supposons qu'une adresse de mémoire contienne 1 octet. Supposons enfin que le système d'exploitation utilise une table de page par programme afin de déterminer où chaque partie du programme se retrouvera en mémoire.

- (a) Quelle sera la taille minimum, en bits, de la table de pages pour un programme?

Solution: La taille de la table de page en nombre de bit = nombre de pages par programme \times nombre de bits pour identifier la page correspondante en mémoire.

- Il y a $2^O/2^M$ pages par programme.
- Il y aura $2^N/2^M$ pages de mémoire. Le nombre de bits pour identifier une page de la mémoire est donc $N - M$. Exemple : Pour une mémoire de 8Ko avec des pages de 1 Ko ($N = 3$ et $M = 0$), il faudra 3 bits (0 à 7) pour identifier les 8 pages possibles.

Réponse : $2^O/2^M \times (N - M)$

- (b) Les adresses logiques à l'intérieur du programme seront sur combien de bits? Les adresses physiques, à l'intérieur de la mémoire seront sur combien de bits?

Solution: Un programme fait 2^O Ko et il y a 1 octet par adresse.
Réponse : $O + 10$ (1 Ko = 2^{10})

- (c) Combien de bits constitueront le champ offset de l'adresse logique?

Solution: Les pages font 2^M Ko et il y a 1 octet par adresse.
Réponse : $M + 10$ (1Ko = 2^{10})

3. Supposons que la mémoire contiennent les processus et les espaces vides suivants:

Si la séquence d'événements suivante survient:

1. P5 est admis dans la mémoire, P5 a une taille de 2ko

Vide, 10ko
P1, 3ko
Vide, 2ko
P2, 5ko
P3, 8ko
Vide, 6ko
P4, 1ko

2. P6 est admis dans la mémoire, P6 a une taille de 5ko
3. P4 est retiré de la mémoire
4. P7 est admis dans la mémoire, P7 a une taille de 1ko

Donnez le contenu de la mémoire après la séquence d'événements si la mémoire est allouée pour des partitions de tailles variables contiguës selon l'algorithme qui suit. Vous pouvez assumer que les adresses de la mémoire vont en augmentant vers le haut dans le schéma ci-haut.

(a) First-Fit (Premier emplacement adéquat)

Solution:	Vide, 5ko
	P6, 5ko
	P1, 3ko
	Vide, 2ko
	P2, 5ko
	P3, 8ko
	Vide, 4ko
	P5, 2ko
	P7, 1ko

(b) Next-Fit (Premier emplacement adéquat après le dernier emplacement choisi)

Solution:	Vide, 4ko
	P7, 1k0
	P6, 5ko
	P1, 3ko
	Vide, 2ko
	P2, 5ko
	P3, 8ko
	Vide, 4ko
	P5, 2ko
	Vide, 1ko

(c) Best-Fit (Le plus petit emplacement adéquat)

Solution:	Vide, 10ko
	P1, 3ko
	P5, 2ko
	P2, 5ko
	P3, 8ko
	Vide, 1ko
	P6, 5ko
	P7, 1ko

4. Quels sont les avantages et les désavantages de l'algorithme First-Fit par rapport à l'algorithme best-fit lorsque l'on parle de stratégie d'allocation de mémoire pour des partitions à taille variable?

Solution: Avantage de First-Fit : Très simple et facile à implémenter

Désavantage de First-Fit : Utilise beaucoup le début de la mémoire

Désavantage de First-Fit : Les fragments de mémoire sont plus petits et plus nombreux

L'avantage compense les désavantages de telle sorte que Best-Fit est rarement utilisé.

5. Qu'est-ce qu'une faute de page?

Solution: Cela survient lorsqu'une page d'un programme n'est pas chargée dans la mémoire et que l'on veuille accéder à cette page.

6. Qu'est-ce que la défragmentation d'un disque dur ou d'une mémoire?

Solution: Un processus qui met tous les espaces de mémoire utilisés l'un à la suite de l'autre afin d'éliminer les trous (fragmentation).

7. Dans quelle condition une page du disque dur sera-t-elle transférée dans la mémoire? Dans quelle condition une page de la mémoire sera transférée vers le disque dur?

Solution: La page du disque dur sera transférée vers la mémoire lorsque le microprocesseur adressera les données ou les instructions qui s'y trouvent. La page de la mémoire sera transférée vers le disque dur si la mémoire est pleine et qu'une page de disque dur doit être mise en mémoire ou si le processus est terminé.

8. Quelles informations sont requise pour traduire une adresse virtuelle/logique en adresse physique lorsque le système d'exploitation alloue de la mémoire pour les processus avec des partitions de taille variable et que tout le processus est contenu dans la partition?

Solution: L'adresse de la mémoire qui correspond à l'adresse 0 du programme (l'adresse de base) et la taille du processus.

9. Quels sont les avantages de séparer la mémoire en page de petite taille qui contiennent une partie des processus seulement par rapport à séparer la mémoire en grande partitions de taille fixe qui contiennent la totalité des processus? Quels sont les désavantages?

Solution: Avantage: Les programmes peuvent être plus gros que la mémoire.

Avantage : Moins de perte de mémoire due à la fragmentation interne.

Avantage : Le nombre et la taille des partitions limitent moins le nombre et la taille des processus pouvant être exécutés.

Désavantage : Beaucoup plus complexe, la translation d'adresse est plus lente ou coûte plus cher en matériel.

Désavantage : Beaucoup plus complexe, il faut de la mémoire pour la translation d'adresse.

10. Pourquoi le registre Program Counter (PC), c'est-à-dire le registre décrivant l'adresse de la prochaine instruction à exécuter contient-il une adresse virtuelle/logique plutôt qu'une adresse physique?

Solution: Toutes les instructions exécutées par le microprocesseur, c'est-à-dire celles du programme, font référence à des adresses virtuelles, c'est-à-dire celles du programme.

11. Les informations nécessaires afin de faire la translation d'adresse sont-elles sauvegardées dans la mémoire ou ailleurs? Quelle composante matérielle ou logicielle de votre ordinateur effectue la translation d'adresse?

Solution: La translation d'adresse doit être très rapide (il faut la faire au moins une fois à toutes les instructions pour aller chercher l'instruction!). Elle est donc effectuée par du matériel spécialisé, le Memory Management Unit (MMU). Le MMU est dans le chipset ou dans le même circuit intégré que le microprocesseur. Il contient des registres ou une petite mémoire cache (comme le TLB) lui permettant de maintenir l'information nécessaire sur chaque processus pour traduire ses adresses. Lorsque la taille des informations nécessaires à la traduction d'adresse est trop grande (la totalité des tables de page par exemple), le gros de l'information est placé en mémoire.