Gestion de la mémoire

RSA : Réseaux et Systèmes Avancés
Deuxième année

★ Exercice 1: Partitions mémoire

On suppose qu'un processus doit être chargé entièrement en mémoire pour être exécuté. Il n' y a pas de mémoire secondaire permettant de faire du swapping. La seule mémoire disponible est une RAM de 256 Ko.

Partitions fixes Soit un système d'allocation mémoire basé sur des partitions fixes et identiques de 64 Ko chacune.

Q 1: Quel est le nombre de programmes maximal que le système peut prendre en charge en même temps ?

Q 2: Supposons l'arrivée, dans cet ordre, des processus A, B, C, D et E ayant comme tailles respectives 5, 25, 65, 32 et 100 Ko. Donner le schéma d'allocation mémoire correspondant. En déduire la quantité de fragmentation interne. Quels sont les inconvénients d'une telle gestion de la mémoire.

Partitions variables On suppose maintenant que l'allocation de la mémoire se fait par partitions de tailles variables (dynamiques).

Q 3: Donner le schéma d'allocation mémoire.

Q 4: Peut-t-on parler de fragmentation interne ici ?

Q 5: On suppose que les processus A et C partent du système avant que le processus F de 70 Ko arrive. Donner le schéma d'allocation mémoire et une solution au problème détecté.

Pagination

Q 6: Expliquer comment la pagination est un mécanisme de partitions fixes qui limite la fragmentation interne. Quelle est la taille moyenne d'une fragmentation interne dans un système paginé ?

Q 7: Donner le schéma d'allocation mémoire qui correspond au scénario ci-dessus. La taille d'une page est de 4 Ko. Calculer la fragmentation interne avant le départ de A et C et comparer la avec celle des partitions fixes.

★ Exercice 2: Algorithme des frères siamois

Soit un système utilisant l'agorithme des frères siamois pour gérer la mémoire. Initialement, la mémoire consiste en un bloc de 256K.

Q 8: Représenter l'état d'occupation de la mémoire après les placements successifs des processus A(5K), B(25K), C(35K), et D(20K) puis les terminaisons successives des processus dans l'ordre A, D, C, et B.

Q 9: Ecrire de façon schématique la fonction trouver-trou(i) qui permet d'allouer un buddy de 2^i et retourne son adresse si possible sinon -1. Le système gère une liste pour chaque puissance de i utilisée et on ne demande pas de détailler la gestion de cette liste. 2^{max} correspond à la taille du plus grand buddy possible.

★ Exercice 3: Tables de pages

On considère une mémoire paginée pour laquelle les cases en mémoire centrale sont de 2 Ko. La mémoire centrale compte au total 9 cases numérotées de 0 à 8. Dans ce contexte, on considère 2 processus A et B de tailles 8 et 16 Ko respectivement.

Q 10: Donner le format d'une adresse logique en explicitant le nombre de bits par champ sachant que l'on se place dans une machine 32 bits.

Q 11: Proposer un schéma de conversion des adresses logiques en adresses physiques de ce système.

Q 12: Quelle est la taille de la mémoire virtuelle ? En déduire le nombre d'entrées dans une table de page de ce système.

Q 13: Donner un contenu possible des tables de pages (case,v/i) des processus A et B ainsi que l'image de la mémoire physique correspondante. On suppose que l'on applique une allocation proportionnelle à la taille des processus A et B. On ne représente dans la table des pages que les entrées effectivement utilisées par chaque processus. On numérote toujours les pages en commençant par 0.

Q 14: Si l'on suppose qu'on utilise une table des pages inversée. Donner le contenu de celle qui correspond au système décrit en gardant la même image de la mémoire physique proposée dans la question précédente.

Q 15: Quel est l'avantage de la table des pages inversée ? Argumenter en s'appuyant sur les données de l'exercice.

Q 16: Quelle est la principale limitation de la table des pages inversée ? Comment peut-t-on l'améliorer ?

★ Exercice 4: Taille de page optimale

Soit un système dans lequel le nombre de lignes du bus d'adresse est n. On suppose que le système utilise la pagination à un seul niveau où p est la taille d'une page. Chacune des entrées de la table des pages a une taille de e octets. Déterminer la valeur optimale de p qui permet de minimiser la fonction qui cumule la perte en mémoire due à la fragmentation interne et au stockage de la table des pages. **A.N.** $n = 32$ et $e = 8$

★ Exercice 5: Pagination multi-niveaux

Soit une machine 32 bits avec une taille de page de 4 Ko.

Q 17: En supposant l'utilisation d'un seul niveau de pagination, quel est le format d'une adresse virtuelle ? En déduire la taille de la table des pages associée à chaque processus.

Q 18: Admettons cette fois-ci que l'on utilise une pagination à 2 niveaux et que pour chaque niveau de pagination, on attribue le même nombre de bits. On garde la même taille de cadre de page soit 4 Ko.
Donner le format de l'adresse virtuelle dans ce cas ainsi que la taille globale des tables de pages.
Quel est l'intérêt de la pagination multi-niveaux ?

★ Exercice 6: Segmentation paginée

Soit une mémoire segmentée paginée pour laquelle les cases en mémoire centrale sont de 4 Ko. La mémoire centrale compte au total 16 cases numérotées de 0 à 15. On considère 2 processus A et B.
Le Processus A a un espace d'adressage composé de trois segments S1A, S2A et S3A qui sont respectivement de 8 Ko, 12 Ko et 4 Ko. Seules les 2 premières pages du segment S1A, la 2ème page du segment S2A et la 1ère page du segment S3A sont chargées en mémoire centrale respectivement dans les cases 3, 12, 1 et 6.
Le processus B a un espace d'adressage composé de deux segments S1B et S2B qui sont respectivement de 16 Ko et 8 Ko. Seules les 2 dernières pages du segment S1B et la 1ère page du segment S2B sont chargées en mémoire centrale respectivement dans les cases 10, 0 et 15.

Remarque. la numérotation (désignation) des pages (comme pour les cases) commence à 0.

Q 19: Représenter sur un dessin la table des segments (taille,segment) et les tables des pages (case,v/i) pour chaque processus ainsi que la mémoire centrale correspondant à l'allocation décrite.

Q 20: Proposer un schéma de MMU capable de faire la conversion des adresses logiques en adresses physiques de ce système.

Q 21: Soit 0x3004, une adresse linéaire pour A, déterminez l'adresses virtuelle correspondantes sous le format (segment,déplacement). Que produit la MMU au final ?

Q 22: Soit 0x832, une adresse linéaire pour B, déterminez l'adresse virtuelle correspondante. Dans ce cas, que génère la MMU ? Quel est le traitement associé ?

★ Exercice 7: Remplacement des pages

On considère un système à mémoire paginée où la taille d'une page est de `200*sizeof(int)`. Un petit processus occupe la page 0 de la mémoire. Il reste 3 cadres de pages libres à l'instant où un programme arrive. Ce dernier initialise le contenu du tableau A défini par :

```
int A[][] = new int[100][100];
```

Q 23: Sachant que ce programme est chargé dans la cadre de page numéro 1 et que les 2 autres pages restantes sont initialement libres, quel est le nombre de défauts de pages expérimenté avec un remplacement LRU et ce pour les deux programmes d'initialisation suivants :

```
for (int j = 0; j < 100; j++)
   for (int i = 0; i < 100; i++)
       A[i][j] = 0;
```

```
for (int i = 0; i < 100; i++)
   for (int j = 0; j < 100; j++)
       A[i][j] = 0;
```