

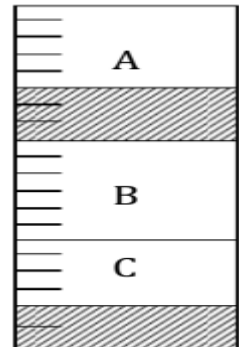
## TD N°04 : Gestion de la Mémoire

### Exercice 4.1 (Questions de Compréhension)

- Q1) Quels sont les deux dispositifs matériels qui permettent au système d'exploitation la protection des processus en exécution ? les décrire brièvement.
- Q2) Dire quelle est la différence principale entre les termes de chacune des paires suivantes :
- A) Une segmentation et une pagination
  - B) Une fragmentation interne et une fragmentation externe
  - C) Un ordonnancement préemptif et un ordonnancement non préemptif
- Q3) Dans un système de gestion mémoire virtuelle à pagination
- A) Quand est-ce qu'un défaut de page se produit
  - B) Dans quel cas est-il nécessaire de réécrire la page sur l'espace de swap
  - C) En déduire son coût en termes entrées/sorties disque
- Q4) Quels sont les inconvénients de l'algorithme de remplacement MFU ?
- Q5) Pourquoi a-t-on intérêt à diminuer le nombre de défauts de pages ?

### Exercice 4.2 (Occupation de la M.C : table de bits et listes chaînées)

Soit le schéma suivant qui montre l'occupation de la M.C en unité (bloc) d'allocation, par les processus A, B, C.



- Q1) Donner les schémas de la table de bits et la liste chaînée correspondantes
- Q2) Dans le cas où le processus B fini son exécution, mettre à jour la table de bits et la liste chaînée pour cette nouvelle situation.

### Exercice 4.3 (Allocation mémoire : First Fit, Best Fit, Worst Fit)

Soit un système qui utilise l'allocation par partitions variables. On parle aussi d'allocation par zones quand la taille des zones allouées est variable, ou d'allocations par blocs quand la taille des blocs alloués est fixe.

On considère à l'instant t l'état suivant de la mémoire centrale :

A	10ko	B	30ko	C	5ko	D	10ko	E	15ko	F	10ko
10ko		20ko		10ko		30ko		5ko		20ko	



Zone libre



Zone occupée

- Q1) Pour les trois algorithmes **First Fit**, **Best Fit** et **Worst Fit** représenter l'évolution de la mémoire centrale en fonction de l'arrivée des événements successifs suivants

- |                                 |                                 |
|---------------------------------|---------------------------------|
| 1. Arrivée du programme G (20k) | 4. Départ du programme E        |
| 2. Départ du programme B        | 5. Arrivée du programme I (40k) |
| 3. Arrivée du programme H (15k) |                                 |

### Exercice 4.4 (Subdivision Binaire)

Simuler la gestion d'une mémoire initialement vide, de taille 1Mo, en utilisant l'allocation par zones siamoises en subdivision binaire (Buddy system binaire) pour les demandes d'allocation (+) et de libération (-) suivantes :

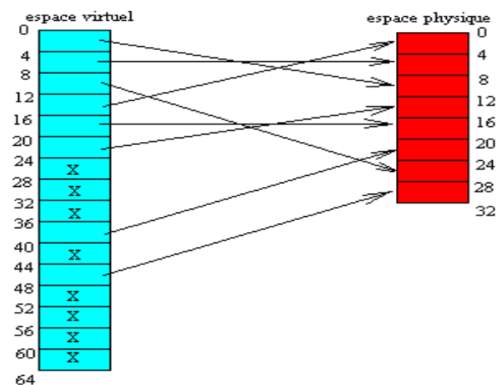
+70K, +35K, +80K, -70K, +60K, -35K, -60K, -80K

### Exercice 4.5 (Translation d'adresse)

Le schéma suivant représente une topographie (mapping) faisant correspondre les adresses virtuelles et les adresses physiques d'un système à mémoire virtuelle (pages de 4Ko). La mémoire physique correspond à 8 pages :

Compléter le tableau de correspondance suivant :

Adresse virtuelle	Adresse physique
20	....
4100	....
....	24684
26000	....



### Exercice 4.6 (Adresses virtuelles)

Supposons que la mémoire physique d'un ordinateur est découpée en 4 pages (de 0 à 3) de 256 octets. C'est-à-dire, la mémoire physique fait 1 Ko ( $256 \times 4 = 1 \text{ Ko}$ ).

Supposons qu'un processus a un espace virtuel de 3 Ko (soit 12 pages, de 0 à 11) et fait successivement référence aux adresses virtuelles suivantes : 249, 255, 300, 500, 540, 600, 650, 700, 750, 800, 900, 1200.

Q1) Trouvez pour chaque adresse virtuelle, le numéro de page virtuelle et le déplacement ?

### Exercice 4.7 (Adresse Physique & Adresse Logique)

Considérons une architecture caractérisée par les hypothèses suivantes :

- Une table de page de taille 128Ko
- Chaque entrée de la table contient une référence vers un cadre de page et un bit de présence/absence.
- La taille d'une page est de 64ko
- La taille de la mémoire physique est de 2Go
- Une adresse virtuelle indexe un octet

Répondez aux questions suivantes en justifiant toujours votre réponse :

- Q1) Combien de cadre de page contient la mémoire physique ?
- Q2) Quelle est la taille en bit d'une entrée de la table de pages ?
- Q3) Quel est le nombre d'entrées dans la table de pages ?
- Q4) Quelle est la taille de la mémoire virtuelle de cette architecture ?
- Q5) Quelle est la taille en bit du bus d'adresse de cette architecture ?

**Q6)** Considérons les deux adresses logiques suivantes exprimées en décimal : 1024 et 65540. Donner si possible les adresses physiques correspondantes (exprimées en décimal) en vous basant sur les 10 premières entrées de la table de pages données ci-dessous.

N° de page	N° de cadre de page	Bit de présence/absence
0	0	1
1	2	0
2	8	0
3	2050	1
4	21054	1
5	31463	1
6	2187	0
7	260	0
8	1266	0
9	1024	1

#### **Exercice 4.8 (Adresse Physique & Adresse Logique)**

On considère un système utilisant la technique de pagination et ayant les caractéristiques suivantes :

- Une table de page ayant  $2^{16}$  entrées
- Chaque entrée de la table de pages est codée sur **16 bits**. Une entrée contient un numéro de cadre de page et un bit de présence/absence.
- Le déplacement (offset) est codé sur **16 bits**
- Une adresse virtuelle indexe **1 octet**

Répondez aux questions suivantes en justifiant toujours votre réponse :

- Q1)** Quelle est la taille d'une page ?
- Q2)** Quelle est la taille de la mémoire physique ?
- Q3)** Quelle est la taille de la mémoire virtuelle ?
- Q4)** Quelle est la taille (en bit) du bus d'adresse de ce système ?
- Q5)** En considérant les huit premières entrées de la table de page représentée par la figure suivante, donner les adresses logiques correspondantes aux adresses physiques 33792 et 66048.

N° de cadre de page	N° de page	Bit de présence/absence
7	0	0
6	0	0
5	0	1
4	1	1
3	0	0
2	0	0
1	2	1
0	3	1

### **Exercice 4.9 (Algorithmes de remplacement de pages)**

Un ordinateur possède une mémoire de 4 pages. Pour chaque page, le gestionnaire de mémoire tient à jour les indicateurs suivants : date de chargement, date de dernière référence, RB (bit indiquant si la page a été référencée), MB (bit indiquant si la page a été modifiée). A un instant donné, la situation est la suivante :

Page	Date de chargement	Date de dernière référence	RB	MB
0	126	279	0	0
1	230	260	1	0
2	120	272	1	1
3	130	280	1	1

**Q1)** Indiquer la page qu'il faudra remplacer prochainement dans le contexte de chacune des stratégies suivantes : **FIFO, LRU, NRU (seconde chance)**

### **Exercice 4.10 (Défauts de Pages)**

On s'intéresse aux systèmes utilisant la pagination

Au cours de son exécution, un programme accède successivement aux pages :

**0, 1, 4, 2, 0, 1, 3, 0, 1, 4, 2, 3.**

Donner la suite des pages présentes en mémoire ainsi que le nombre de défauts de pages pour chacun des cas suivants :

- Q1)** Si on utilise l'algorithme **FIFO** et sachant que le système alloue à ce programme un espace de **3 pages**.
- Q2)** Si on utilise l'algorithme **LRU** et sachant que le système alloue à ce programme un espace de **3 pages**.
- Q3)** Si on utilise l'algorithme **Optimal** et sachant que le système alloue à ce programme un espace de **3 pages**.
- Q4)** Si on utilise l'algorithme **FIFO** et sachant que le système alloue à ce programme un espace de **4 pages**.
- Q5)** Si on utilise l'algorithme **LRU** et sachant que le système alloue à ce programme un espace de **4 pages**.

### **Exercice 4.11 (Défauts de Pages)**

Un programme a un espace virtuel de 600 mots. On considère la suite des adresses virtuelles : **34, 123, 145, 510, 456, 345, 412, 10, 14, 12, 234, 336, 412.**

- Q1)** Donner la suite des numéros de pages référencés, sachant qu'elles comportent **100 mots**.
- Q2)** Le programme dispose de 300 mots en mémoire centrale. Calculer le taux de défauts de page (en supposant la mémoire initialement vide) pour les algorithmes **OPT (Optimal), FIFO (First Page In, First Page Out), LRU (Least Recently Used), FIFO de la seconde chance.**