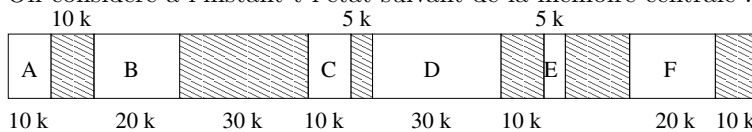


# Mémoire

## 1 Allocation par partition

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



Représenter l'évolution de la mémoire (pour les algorithmes First Fit, Best Fit et Worst Fit) en fonction de l'arrivée des évènements suivants :

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

## 2 Pagination

Pour chacune des adresses virtuelles suivantes, donnez le numéro de page virtuelle et le déplacement pour des pages de 4 ko et de 8 ko : 0xbffeb5a9, 0x0804fc02, 0x518c0a51

## 3 Pagination (suite)

On considère un système de pagination à 2 niveaux :

- Les adresses (virtuelles et physiques) sont codées sur 32 bits
- Les 10 premiers bits d'une adresse virtuelle forment le premier index, les 10 suivants forment le second index, et les 12 restant forment le déplacement
- On suppose que chacune des entrées de ces tables occupe 32 bits.

Combien peut-il y avoir de cadres de pages au maximum ?

Quelle taille feront ces cadres de pages ?

Combien d'espace mémoire utilisera-t-on pour gérer la table des pages d'un processus ?

## 4 Remplacement de page

Soit la liste des pages virtuelles référencées aux instants  $t = 1, 2, \dots, 11$  :

3 5 6 8 3 9 6 12 3 6 10

La mémoire est composée de 4 cases initialement vides.

Représentez l'évolution de la mémoire au fur et à mesure des accès pour chacune des deux politiques de remplacement de pages FIFO et LRU (*Last Recently Used*). Notez les défauts de pages éventuels.

## 5 Temps d'accès effectif

Sur un système qui a recours à la mémoire paginée à la demande, il faut 200 ns pour satisfaire une requête mémoire si la page reste en mémoire. Si tel n'est pas le cas, la requête prend 7 ms si un cache libre est disponible ou si la page à extraire n'a pas été modifiée. Il faut par contre 15 ms si la page à extraire a été modifiée.

Quel est le temps d'accès effectif si le taux de défaut de page est de 5 % et que, 60 % du temps, la page à remplacer a été modifiée ?

## 6 Mémoire segmentée

On considère la table de segments suivante :

Segment	Base	Longueur
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

Calculez les adresses physiques des adresses logiques suivantes : (0, 430) (1, 10) (1, 11) (2, 500) (3, 400) (4, 112)

## 7 Mémoire segmentée (suite)

On considère une mémoire segmentée paginée pour laquelle les cases en mémoire centrale sont de 4 Ko. La mémoire centrale compte au total 15 cases numérotées de 1 à 15. Dans ce contexte, on considère deux 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.

Le processus B a un espace d'adressage composé de deux segments S1B et S2B qui sont respectivement de 16 Ko et 8 Ko.

Pour le processus A, seules les pages 1 et 2 du segment S1A, la page 2 du segment S2A et la page 1 du segment S3A sont chargées en mémoire centrale respectivement dans les cases 4, 5, 10 et 6.

Pour le processus B, seules les pages 2 et 3 du segment S1B et la page 1 du segment S2B sont chargées en mémoire centrale respectivement dans les cases 11, 2 et 15.

Représentez sur un dessin les structures allouées (table des segments, tables des pages) et la mémoire centrale correspondant à l'allocation décrite.

Si 4098 et 12292 sont des adresses linéaires pour A, déterminez les adresses virtuelles et réelles correspondantes.

Même question avec 16389 pour A et 8212 pour B

## 8 Mémoire paginée à la demande

Dans ce problème, on utilise des adresses décimales, une taille de page de 2000 octets et la table des pages suivante :

Page	Adresses	In/Out	Cadre
0	0 - 1999	In	20
1	2000 - 3999	Out	22
2	4000 - 5999	In	200
3	6000 - 7999	In	150
4	8000 - 9999	Out	30
5	10000 - 11999	Out	50
6	12000 - 13999	In	120
7	14000 - 15999	In	101

Parmi les adresses virtuelles suivantes, laquelle génère un défaut de page? Pour celles qui ne génèrent pas de défaut de page, quelle est leur adresse physique après translation?

1. 10451
2. 5421
3. 14123
4. 9156