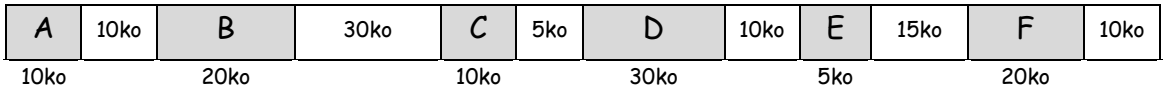
TD N°04 : Gestion de la Mémoire (Supplémentaire)

Exercice 4.1 (Allocation de mémoire)

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



Zone libre — Zone occupée

Des requêtes d'allocation de mémoire arrivent dans l'ordre suivant : G (20 Ko), H (10 Ko), I (5 Ko) et K (25 Ko).

Q1) A quelles adresses sont alloués les blocs si on utilise la politique **First Fit** ? Quel décision prise vous semble inefficace ?

Q2) Même question avec la politique Best Fit

Q3) Même question avec la politique Worst Fit

Exercice 4.2 (Allocation de mémoire)

Dans un système de gestion mémoire à partitions variables, on constate que la liste des "trous" est la suivante (dans l'ordre des adresses mémoire croissantes) :

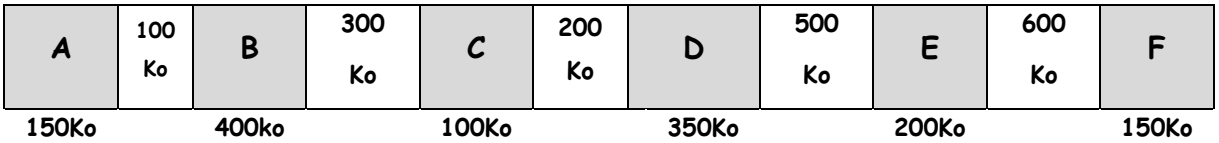
10K 4K 20K 18K 7K 9K 12K et 15K

On veut placer successivement des données de volumes respectifs : **12K 10K 9K** dans la mémoire.

Q1) Indiquer, dans l'ordre des adresses croissantes, la nouvelle liste des trous après l'opération précédente dans le cadre de chacune des stratégies de placement : **First Fit, Best Fit, Next Fit** et **Worst Fit.**

Exercice 4.13

Soit un système d'exploitation, qui utilise l'allocation contiguë par partition variable. On considère à l'instant T l'état suivant de la mémoire centrale (les zones hachurées sont libres) :



Zone Libre — Zone Occupée

Q1) Représenter l'évolution de la mémoire centrale, en fonction de l'arrivée des processus K, L, M et O, selon cet ordre : K (200 Ko), L (450 Ko), M (250 Ko) et O (50 Ko) :

A) En utilisant la stratégie d'allocation **First Fit**

B) En utilisant la stratégie d'allocation **Best Fit**

Q2) Nous désirons choisir un moyen de mémorisation des parties libres et parties occupées de la mémoire, ainsi qu'elle était représentée avant l'arrivée des nouveaux processus.

A) Donner la table de bits correspondant à la répartition précédente, en supposant que l'unité d'allocation est le 50Ko.

B) Soit une représentation des parties libres/occupées sous la forme d'une liste chaînée, dont chaque nœud contient un bit d'état (L/O), la taille de la partition, et un pointeur sur la prochaine partition. Représenter la liste chaînée correspondant à la répartition précédente (unité d'allocation : 50Ko).

C) Nous désirons charger le processus P (400Ko) dans la mémoire selon la stratégie d'allocation **First Fit**.

i. Représenter l'état de la mémoire suite au chargement de P.

ii. Décrire les étapes nécessaires pour localiser l'emplacement adéquat du nouveau processus dans la mémoire en utilisant :

1. La table de bits
2. La liste chaînée

iii. Comment proposez-vous de modifier la structure de la liste chaînée de manière à rendre la mise à jour de la mémoire encore plus rapide ?

Exercice 4.4 (Ordonnancement & gestion de mémoire)

On considère un système disposant de **16 Ko** de mémoire centrale dont **4 Ko** sont occupés par la partie résidente du système d'exploitation. Nous supposons que :

- La mémoire centrale utilise un partitionnement variable,
- La mémoire centrale utilise une stratégie de placement se basant sur l'algorithme premier ajustement ou **First Fit**,
- L'ordonnancement sur la **mémoire** se fait selon **FIFO**,
- L'ordonnancement sur la **CPU** se fait selon l'algorithme du tourniquet (**Round Robin**) avec un quantum **q = 3ms**,
- Chaque processus effectue un calcul suivi d'une **E/S** en utilisant son propre périphérique,
- Les processus arrivent conformément au tableau suivant :

Processus	Instant (t)	Taille (Ko)	Temps CPU (ms)	Durée E/S (ms)
A	0	3	9	2
B	4	5	6	9
C	6	5	4	4
D	8	4	2	6
E	10	1	4	3
F	12	1	5	1
G	16	1	3	2
H	18	3	3	8

Q1) Donnez le diagramme de GANT ainsi que les états d'occupation de la mémoire aux différentes étapes de traitement de ces processus.

Exercice 4.5 (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.

Q1) 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 ?

Exercice 4.6 (Segmentation Paginée, Pagination à deux niveaux)

Considérez un système de gestion de mémoire qui a les caractéristiques suivantes :

- Un adressage virtuel sur **32 bits** (dont **14 bits** pour les numéros de segments et **6 bits** pour les numéros de pages)
- Une taille de Page de **4Ko**
- Une mémoire physique de **1 Mo**

Supposez que le système utilise la segmentation paginée

Q1) Quelle est la taille du plus grand segment (**en pages/Ko**) ?

Q2) Quelles sont les données manquantes à ce problème pour traduire l'adresse virtuelle de **32 bits** suivante : **0xAE854C9C** en adresse physique ?

Si vous aviez ces informations, identifiez brièvement les étapes à suivre pour effectuer cette translation.

Supposons maintenant que le système considéré utilise une pagination à deux niveaux, où les entrées du de la table de page de premier niveau sont sur 4 octets

- La structure de l'adresse virtuelle est composée de

#page niveau 1 (10 bits)	#page niveau 2 (10 bits)	Déplacement page (12 bits)

Q3) Si un processus utilise tout l'espace adressable qui lui est fourni, combien de pages seront-elles nécessaires pour contenir toutes les tables de pages de ce processus

Q4) Un second processus nécessite 22Mo pour s'exécuter entièrement (son code, ses données, pile…). La partie contenant son code est disposée dans sa mémoire virtuelle aux adresses suivantes [2Mo à 6Mo-1], les données sont quant à elles dans l'intervalle [12Mo à 21Mo-1]. Si nous devons charger les tables de pages associées à ces deux parties, combien de pages de niveaux 2 seront chargées en mémoire centrale

Q5) Supposez maintenant que le système utilise une pagination pure et que les bits **12 à 31** correspondent à un numéro de page. Si nous utilisons une table inversée, combien d'entrées la table de pages inversée contiendra-t-elle ?

Exercice 4.7 (Adresse Physique & Adresse Logique)

Etant donnés :

- Une table des pages de taille **128 Ko**
- Le nombre d'entrées de la table des pages égal à **65536**
- Le déplacement est codé sur **16 bits**
- Une entrée de la table des pages est de la forme : **|n|1|3|** où
  - ✓ **n** est nombre de bits pour coder un cadre de page (une case)
  - ✓ **1** est le bit d'absence/présence
  - ✓ **3** est le nombre de bits pour coder la date de chargement de la page
  - ➔ Chaque entrée de la table contient donc **n+4 bits**

On vous demande de (vous devez détailler votre réponse) :

Q1) Déterminer la valeur de **n**

Q2) Calculer la taille de la mémoire physique.

Q3) Donner la taille de l'espace d'adressage virtuel

Q4) Donner le nombre de bits du bus d'adressage.

Q5) Dire si le nombre d'entrées de la table des pages change si on augmente la taille de la mémoire physique de **1 Mo**. Si oui, de combien augmente-il ?