

# Segmentation

**Attention :** Dans ce TD, toutes les *tailles*, *adresses* et *numeros de segments* sont donnés en hexadécimal.

**Exercice 1. Questions de cours** Les questions de cours sont à destinées à vous permettre de vérifier votre compréhension du cours. Elles sont à travailler à l'avance et ne seront pas traitées en TD ou TP.

1. Expliquez ce qu'est l'édition de liens (en quelques mots).
2. Quelle est la meilleure stratégie d'allocation dans le cas d'une allocation contiguë ?
3. Donnez deux avantages de la segmentation.
4. Comment est déterminée une erreur de segmentation ?

**Exercice 2. Mémoire contiguë** On considère un système muni de 64ko de mémoire et capable de gérer plusieurs processus en mémoire simultanément.

1. Quelle est la taille de l'adresse physique ?
2. Quelle est la taille de l'adresse logique ?

On décide d'utiliser un algorithme d'allocation contiguë qui met en attente les processus lorsqu'il n'y a plus de bloc libre de taille suffisante. On considère l'exécution suivante, pour laquelle nous donnons pour chaque processus sa date d'arrivée, son besoin en mémoire (donné à la création du processus) et sa durée d'exécution totale :

Processus	Arrivée	Mémoire	Durée
P1	0	16ko	10
P2	2	8ko	4
P3	2	32ko	12
P4	4	12ko	12
P5	8	4ko	6
P6	12	32ko	4

3. On suppose que l'allocation se fait suivant un algorithme de type "First Fit". Décrivez précisément l'allocation mémoire à chaque pas de temps, en expliquant au fur et à mesure comment sont positionnés les processus.
4. Quel est le temps total d'exécution ?
5. Le taux de fragmentation est la quantité de mémoire dans les "trous" de mémoire libre ramenée à la quantité de mémoire actuellement allouée. Quel est le taux de fragmentation au temps  $t_8$ ,  $t_{10}$  et  $t_{12}$  ?
6. L'exécution serait-elle meilleure avec un algorithme de type "Best Fit" ou "Worst Fit" ?
7. Quelle serait l'allocation optimale afin de minimiser le temps de calcul sur cet exemple ?

**Exercice 3. Segmentation simple** On considère un système de gestion de la mémoire utilisant la segmentation avec des adresses logiques sur 16bits dont les 4bits de poids fort indiquent le numéro de segment. Les premières lignes de la table des segments d'un processus sont données dans le tableau suivant :

Segment	Base	Limite
0	02B9	5FF
1	2A00	014
2	0090	100
3	C3A7	5D0
4	1D52	09F

1. Quelle quantité de mémoire un processus peut-il utiliser au maximum ?
2. Quelle est la taille maximale d'un segment ?
3. Quelles sont les adresses physiques correspondant aux adresses logiques suivantes : 0x0430, 0x1010, 0x2B0F, 0x34FF et 0x4100 ?
4. En supposant que toutes les autres lignes de la table ont les valeurs base et limite à zéro, quelle quantité de mémoire est allouée à ce processus.

**Exercice 4. Mini segmentation** On considère un système de gestion de la mémoire pour un micro-contrôleur, c'est-à-dire un très petit processeur consommant très peu d'énergie dont le prix est très bas. Ce système étant très simple, ses capacités sont limitées. Il dispose quand même d'une gestion de la mémoire segmentée mais un peu particulière.

Ce système dispose de 16ko de mémoire physique. Afin de réduire la consommation mémoire, une seule table des segments est utilisée pour l'ensemble des processus. Cette table permet de décrire 128 segments, chaque ligne est composée de :

- 3bits indiquant le numéro du processus à qui appartient le segment décrit (le numéro 7 indiquant un segment partagé par tous les processus) ;
- 9bits indiquant l'adresse de base divisée par 32 (les adresses de base sont des multiples de 32 la division est donc exacte).
- 4bits donnant la taille du segment divisée par 64 (les segments ont forcément une taille multiple de 64 la division est donc exacte) ;

Cette manière de faire la segmentation permet une gestion de la mémoire offrant une protection et flexibilité tout à fait satisfaisantes pour ce type de processeur en ne consommant que 256o de la faible quantité de mémoire disponible.

*Les multiplications par 32 et 64 sont très simples à réaliser en binaire simplement en ajoutant respectivement 5 et 6 zéros à la fin du nombre.*

1. Quelle est la taille de l'adresse physique ?
2. Quelle est la taille maximale d'un segment ?
3. Quelle est la taille et la composition de l'adresse logique ?  
Un extrait de la table des segments est donné ci-dessous ;

Segment	Processus	Base	Limite
C7	3	103	2
03	1	1CC	3
17	1	0E0	1
E5	2	000	A
06	7	0E5	4
2A	4	1B0	3

4. Donnez, pour le processus 3, les adresses physiques correspondant aux adresses logiques suivantes : A8C7 et 1A2C.

**Exercice 5. Segmentation avancée** On considère un système doté de 1Mo de mémoire physique utilisant la segmentation. Chaque processus peut utiliser un maximum de 256 segments d'une taille maximale de 64ko. 128 de ces segments sont locaux au processus et sont décrits par une table des segments propre à ce processus. Les 128 autres segments sont communs à tous les processus et décrits par une table des segments commune à tous les processus du système. Le bit de poids fort de l'adresse logique indique, quand il vaut 1, que le segment référencé par cette adresse logique est un segment global.

Un segment ne peut commencer qu'à une adresse multiple de 256, cela implique que le dernier octet de son adresse de base sera toujours 0 et qu'il n'est donc pas nécessaire de le stocker dans la table des segments.

Tous les segments, qu'ils soient locaux ou globaux, sont décrits dans une des deux tables par leur adresse de base, leur limite ainsi que par 3bits de protection R, W et X, indiquant si un accès en lecture, en écriture ou en exécution est autorisé, ainsi qu'un bit S indiquant si l'accès au segment est réservé au mode superviseur du système.

1. Quelle est la taille des adresses physique ?
2. Quelle est la taille de l'adresse logique ?
3. Quelle est la taille de la table des segments locale d'un processus ?

On suppose qu'un processus P1 est présent dans le système, on donne ci-dessous des extraits des tables des segments. Vous pouvez considérer que toutes les lignes qui ne sont pas présentes sont entièrement à zéro. (la colonne RWXS indique la valeur des 4bits en notation binaire)


Table locale de P1

Seg.	Base	Limite	RWXS
00	000	0000	0000
0C	020	4000	1010
13	1D4	0800	1000
21	000	0400	0000
2A	C70	D704	1100
48	1D4	0800	1101

Table globale

Seg.	Base	Limite	RWXS
21	000	0200	1100
2A	A5C	0100	1100
6C	032	1000	0010

4. Que se passe-t-il lorsque le processus réalise des accès en écriture aux adresses suivantes : 2A04E5, A10386 et 0c1F5E ?
5. Où est probablement stocké le code du processus P1 ?

 Les questions suivantes sont plus générales et sont destinées à vous amener à réfléchir sur les possibilités offertes par ce type de système de gestion de la mémoire. La conception de ces systèmes est toujours un compromis entre les contraintes pratiques et les fonctionnalités offertes.

6. Qu'y a-t-il de particulier avec les segments locaux 13 et 48 ? À quoi sert ce type de configuration ?
7. Qu'y a-t-il de particulier avec le segment local 00 ? À quoi sert-il ?
8. En quoi ce système de gestion de la mémoire est-il particulièrement intéressant pour l'OS ?
9. Le fabricant de ce système souhaite produire une nouvelle version disposant de 4Mo de mémoire, quelle(s) modification(s) proposeriez-vous pour supporter cette quantité de mémoire physique ?