

TD n°6 : Gestion de la mémoire

CORRECTION

Exercice 1 – Notions de cours

- 1) cf. cours
- 2) fragmentation interne → système paginés, fragmentation externe → systèmes segmentés

Exercice 2 – Segmentation

1) L'adresse physique s'obtient en ajoutant l'adresse de base du segment au déplacement dans le segment, mais à condition que le déplacement ne soit pas supérieur à la taille du segment moins 1 (on compte le déplacement en partant de 0) :

- (0:128) : déplacement valide ($128 < 234$). $\text{Adr_physique} = \text{base} + \text{limite} = 540 + 128 = \mathbf{668}$.
- (1:100) : déplacement valide ($100 < 128$). $\text{Adr_physique} = \text{base} + \text{limite} = 1254 + 100 = \mathbf{1354}$.
- (2:465) : déplacement invalide ($465 > 328$).
- (3:888) : déplacement valide ($888 < 1024$). $\text{Adr_physique} = \text{base} + \text{limite} = 2048 + 888 = \mathbf{2936}$.
- (4:100) : déplacement valide ($100 < 200$). $\text{Adr_physique} = \text{base} + \text{limite} = 976 + 100 = \mathbf{1076}$.
- (4:344) : déplacement invalide car ($344 > 200$).

2) Non. Dans un segment de longueur 200, les déplacements valides sont dans l'intervalle [0-199].

Exercice 3 – Pagination

1) L'espace d'adressage du processus est l'espace d'adressage virtuel formé par les pages. Comme il y a 8 pages, la taille de l'espace virtuel est de $8 * 256 = 2048$ mots.

2) Comme les cadres sont numérotés sur 3 bits, il y a $2^3 = 8$ cadres. Taille d'un cadre = taille d'une page donc la mémoire physique comporte $8 * 256 = 2048$ mots (= 2Ko).

3) La conversion d'une adresse virtuelle en adresse réelle est réalisée de la façon suivante :

(a) Calcul du numéro de la page et du déplacement dans la page.

(b) Recherche dans la table de pages de l'entrée qui correspond à la page de façon à en déduire le numéro du cadre.

(c) L'adresse physique (réelle) est obtenue en ajoutant le déplacement à l'adresse physique de début du cadre.

Voici le détail des calculs pour les adresses demandées :

- $240 = 0 * 256 + 240 \rightarrow$ page = 0 et déplacement = 240. D'après la table des pages, cadre = 3.

D'où $\text{Adr_phys} = 3 * 256 + 240 = 1008$

- $546 = 2 * 256 + 34 \rightarrow$ page = 0 et déplacement = 34. D'après la table des pages, cadre = 0.

D'où $\text{Adr_phys} = 0 * 256 + 34 = 34$.

- $1578 = 6 * 256 + 42 \rightarrow$ page = 6 et déplacement = 42. D'après la table des pages, cadre = 5.

D'où $\text{Adr_phys} = 5 * 256 + 42 = 1322$.

- 2072 est en dehors de l'espace d'adressage virtuel du processus (2048 mots).

4) $770 = 3 \times 256 + 2$. Il s'agit d'une adresse située dans la page 3. Or d'après la table des pages, cette page n'est pas présente en mémoire. Une référence à cette adresse provoquera donc un défaut de page.

5) D'après la table de pages, cette page se trouve dans le cadre 010. L'adresse physique s'obtient donc simplement en substituant aux 4 bits de poids fort de l'adresse virtuelle les 3 bits du numéro de cadre : 010 0000 0000 0111.

Exercice 4 – Segmentation paginée

Segments	Pages	Cadres
S1	2	2
	3	0
S2	2	9
S3	1	12

$8212 = 2 \times 4096 + 20$

- 1) segment = S1
- 2) page = 3
- 3) déplacement = 20
- 4) cadre = 0
- 5) déplacement = 20
- 6) L'adresse physique est donc 20 en décimal. L'adresse physique est exprimée sur 16 bits ($64\text{Ko} = 2^{16}$), dont 4 bits pour le numéro de cadre et 12 bits pour le déplacement dans la case ($4\text{Ko} = 2^{12}$), ce qui nous donne en binaire 0000 0000 0001 0100.

Exercice 5 – Algorithmes de remplacement de pages

1) Il suffit de diviser l'adresse par 100, ce qui donne : 0, 1, 1, 5, 4, 3, 4, 0, 0, 0, 2, 3, 4.

2)
OPT :

Req	0	1	1	5	4	3	4	0	0	0	2	3	4
C ₁	0	0	0	0	0	0	0	0	0	0	2	2	2
C ₂	-	1	1	1	4	4	4	4	4	4	4	4	4
C ₃	-	-	-	5	5	3	3	3	3	3	3	3	3
Def	X	X		X	X	X					X		

FIFO :

Req	0	1	1	5	4	3	4	0	0	0	2	3	4
C ₁	0	0	0	0	4	4	4	4	4	4	2	2	2
C ₂	-	1	1	1	1	3	3	3	3	3	3	3	4
C ₃	-	-	-	5	5	5	5	0	0	0	0	0	0
Def	X	X		X	X	X		X			X		X

LRU :

Req	0	1	1	5	4	3	4	0	0	0	2	3	4
C ₁	0	0	0	0	4	4	4	4	4	4	4	3	3
C ₂	-	1	1	1	1	3	3	3	3	3	2	2	2
C ₃	-	-	-	5	5	5	5	0	0	0	0	0	4
Def	X	X		X	X	X		X			X	X	X

FINUFO :

Req	0	1	1	5	4	3	4	0	0	0	2	3	4
C ₁	0 ₁	0 ₁	0 ₁	0 ₁	4 ₁	4 ₁	4 ₁	4 ₁	4 ₁	4 ₁	2 ₁	2 ₁	2 ₁
C ₂		1 ₁	1 ₁	1 ₁	1 ₀	3 ₁	3 ₁	3 ₁	3 ₁	3 ₁	3 ₀	3 ₁	3 ₀
C ₃				5 ₁	5 ₀	5 ₀	5 ₀	0 ₁	0 ₁	0 ₁	0 ₀	0 ₀	4 ₁
Def	X	X		X	X	X		X			X		X

Exercice 6 – Temps d'accès

1)

a) Dans l'algorithme LRU, on retire la page la moins récemment utilisée. Il s'agit donc de choisir une page selon le critère de la colonne $T_{\text{dernier accès}}$. La page à retirer est celle chargée dans le **cadre 1**, qui a été accédée au temps 255.

b) Dans l'algorithme FIFO, on retire la page qui est en mémoire depuis le plus longtemps. Il s'agit donc de suivre le critère de la colonne $T_{\text{chargement}}$. La page à retirer est celle chargée dans le **cadre 2** qui est en mémoire depuis le temps 110.

c) Dans l'algorithme FINUFO, on retire la page qui est en mémoire depuis le plus longtemps, donc selon le critère de la colonne $T_{\text{chargement}}$, sauf si son bit R est à 1, auquel cas on le remet à 0 et on poursuit la recherche dans l'ordre. Dans l'exemple, la page chargée dans le cadre 2 est la plus ancienne, mais son bit R est à 1. La suivante dans l'ordre est la page chargée dans le **cadre 0** dont le bit R est à 0. C'est donc celle qui est choisie.

d) Dans l'algorithme NRU, les pages sont séparées en deux catégories basées sur les valeurs des bits R et M :

Classe 0 : R=0, M=0

Classe 1 : R=0, M=1

Classe 2 : R=1, M=0

Classe 3 : R=1, M=1

On retire une page au hasard dans la classe la plus basse non-vidée. Il s'agit donc de retirer la page du **cadre 0**, qui appartient à la classe 0.

2) Le taux d'accès réalisés en 100 ns est de 65%. Parmi les 35% accès menant aux défauts de page, 70% ont besoin de 20 ms et le reste de 30% ont besoin de 10 ms.

$$T_{\text{accès moyen}} = 0.65 * 0.0001 + 0.35 * (0.7 * 20 + 0.3 * 10) = \mathbf{5.950065 \text{ ms}}$$