

Pagination

Attention : Dans ce TD, toutes les *adresses* et *portions d'adresses* sont donnees en hexadecimal.

Exercice 1. Questions de cours Les questions de cours sont a destinees a vous permettre de verifier votre comprehension du cours. Elles sont a travailler a l'avance et ne seront pas traitees en TD ou TP.

1. En quoi consiste la pagination de la memoire ?
2. Quel est le principal probleme de la pagination ?
3. Quel est l'interet de la pagination hierarchique ?
4. Donnez deux avantages de la memoire virtuelle.

Exercice 2. Fondements On suppose un espace d'adresses logiques de 8 pages de 1024 octets chacune permettant d'accéder a une memoire physique de 32 cadres de pages.

1. Expliquez pourquoi les tailles de pages sont toujours une puissance de 2.
2. Combien de bits comporte l'adresse logique ? L'adresse physique ?
3. On suppose maintenant que le systeme dispose de 2048ko de memoire logique organise avec des pages de 8ko. Decrivez le systeme d'adressage logique.
4. Quelle est la taille maximum de la table des pages ?
5. On suppose que, dans le systeme de la question precedente, on a trois processus qui s'exécutent sur le systeme : P1 necessitant 200ko, P2 de 545ko et P3 de 337ko. Quelle est la quantite de memoire reellement utilisee par l'exécution de ces trois processus ? Quel est le taux de fragmentation ?

Exercice 3. Pagination a 1 niveau On considere un systeme 32bits utilisant la technique de pagination avec des cadres de pages de 64ko. Chaque processus peut utiliser au plus 4Go de memoire virtuelle et le systeme peut accueillir jusqu'a 2048 processus.

1. Quelle est la taille maximale de la memoire physique ?
2. Quelle est la taille de l'adresse logique ?
3. Quelle est la quantite maximale de memoire virtuelle utilisee par le systeme ?
4. On suppose que le systeme dispose de 1Go de memoire physique. Quelle est la taille de l'adresse physique ?
5. Sur combien de bits est code le decalage dans l'adresse logique ? Dans l'adresse physique ?
6. Sur combien de bits est code le numero de page dans l'adresse logique ?
7. Sur combien de bits est code le numero de cadre dans l'adresse physique ?
8. Quelle est la taille maximale de la table des pages d'un processus ?
9. En considerant les huit premieres entrees de la table de page donnee ci-apres, calculez les adresses physiques correspondant aux adresses logiques 00030B72 et 00060D81 ?

Page	Cadre	Validite
0	000	0
1	010	0
2	000	1
3	0B3	1
4	2A0	0
5	09B	0
6	0F0	1
7	2DD	1

10. A l'inverse, pouvez-vous indiquer si la donnee situee a l'adresse physique 2A0DE37 appartient au processus ? Expliquez.
11. Est-ce que 37 est une adresse physique valide ? Et 90003145 ? Expliquez.

Exercice 4. Pagination à 1 niveau On considère un système disposant de 4Mo de mémoire physique utilisant la technique de la pagination à 1 niveau avec des cadres de page de 1ko. Chaque processus dispose de 16Mo de mémoire virtuelle.

1. Quelle est la taille des adresses physiques ?
2. Sur combien de bits est codé le numéro de page dans l'adresse physique ?
3. Quelle est la taille des adresses logiques ?
4. Combien de lignes y a-t-il dans la table des pages d'un processus ?

Le tableau suivant donne un extrait de la table des pages d'un processus :

Page	Cadre	Validité
AC2	000	0
AC3	35E	1
AC4	12A	1
AC5	000	0
AC6	000	0
AC7	000	0
AC8	000	0
AC9	1A7	0

5. Calculez les adresses physiques correspondant aux adresses logiques 2B0E4C et 2B27F0.
6. Calculez l'adresse logique que le processus doit utiliser afin de réaliser un accès mémoire à l'adresse physique 4A8F1.
7. Que doit faire l'OS pour que le processus puisse réaliser un accès mémoire à l'adresse physique D12A ?

Exercice 5. Pagination à 2 niveaux On se place dans un système de mémoire de 4Go de mémoire géré de manière paginée avec des cadres de page de 4Ko. Chaque processus peut utiliser jusqu'à 64Mo de mémoire. Le système d'exploitation autorise jusqu'à 1024 processus.

1. Quelle est la taille (en bits) de l'adresse logique
2. Quelle est la taille (en bits) de l'adresse physique
3. Combien y a-t-il de cadres de page dans la RAM ?
4. Combien chaque processus peut-il contenir de pages ?
5. On suppose que la pagination se fait sur un seul niveau. Quelle quantité de mémoire est consommée par les tables des pages ?
6. On suppose que la pagination se fait sur deux niveaux avec un répertoire sur 6 bits. Quelle quantité de mémoire est consommée par un processus qui utilise les plages d'adresses logiques suivantes ?
 - de 0 00 1B 07 à 0 00 2C 08
 - de 0 03 CA 00 à 0 0E 00 00
 - de 0 0D AB 10 à 0 0F B1 CC
 - de 0 2B 10 21 à 0 2B 3A 72

Exercice 6. Pagination à 2 niveaux On considère un système de gestion de mémoire paginée à deux niveaux tel que :

- Les adresses virtuelles et physiques sont toutes deux codées sur 32bits ;
- Les 10 premiers bits de l'adresse virtuelle forment le premier index ($n_1 = 10$), les 10 bits suivant forment le deuxième index ($n_2 = 10$) et les 12 bits restant le déplacement ($m = 12$) ;
- Les tables de pages et le répertoire de chaque processus sont stockées dans la RAM et chaque entrée correspond à une adresse différente dans la RAM.

1. Sur combien d'octets seront codées les données du répertoire ? Quelle est la taille maximum du répertoire ?
2. Sur combien d'octets seront codées les données d'une table de page ? Quelle est la taille maximum d'une table de pages ?
3. Quelle est la quantité maximale de mémoire nécessaire pour stocker toutes ces tables ?
4. On considère un processus avec le répertoire suivant :

Index	Table
0	0A7C 0002
1	0A7C 006E
2	0A7C 0072
3	0A7C 005A

Et on souhaite résoudre l'adresse logique: 00C02DF6. Quelle table est concernée ?

5. On suppose que la table des pages concernée (par le résultat de la question précédente) commence par les entrées suivantes :

Page	Cadre	Validité
0	17AC0	0
1	02BCF	1
2	07F93	1
3	15BB0	1

Donnez l'adresse physique correspondant à l'adresse logique 00C0 2DF6.

Exercice 7. Segmentation paginée On considère un système muni de 64 ko de mémoire physique gérée de manière segmentée et paginée. Chaque processus peut utiliser 16 segments de 1 ko et le système supporte jusqu'à 256 processus. Les cadres de page font 512 o.

1. Quelle est la taille de l'adresse physique ?
2. Quelle est la taille de l'adresse logique ?
3. Quelle est la taille maximale de la mémoire virtuelle ?
4. Rappeler ce qu'est un segment global.
5. Dans cette question, on suppose que la moitié des segments sont globaux. Dans ce contexte, quelle est la taille de la mémoire virtuelle ?
6. Combien de pages un processus peut-il utiliser au maximum ?
7. On rappelle que l'adresse linéaire doit permettre d'adresser toute la mémoire d'un processus, mais qu'elle n'a pas besoin d'adresser l'ensemble de la mémoire virtuelle. Quelle est la taille et la composition de l'adresse linéaire ?
8. Quelle est la taille (en nombre de bits) de chaque ligne de la table des descripteurs lorsque tous les segments sont locaux ?
9. Même question si la moitié des segments sont globaux (ce qui est la configuration habituelle sur un OS moderne).

À un moment de l'exécution, plusieurs processus P_1, P_2, \dots, P_N sont en exécution dans le système. L'état du système est partiellement décrit ci-après :

Table des segments de P_1 :

Segment	Limit	Base
00	085	0000
01	3B6	3000
02	341	2000
03	225	0086
04	05F	1B80

Table des segments de P_2 :

Segment	Limit	Base
00	0A5	1001
01	107	0C00
02	3B6	3000
03	0A3	01CF

Table des pages de P_1 :

Page	Cadre	Valide
00	36	0
01	7A	0
02	32	1
0E	00	1
0F	2A	0
10	2B	0
11	14	1
18	6C	1
19	55	1
1A	31	1
1B	30	0

Table des pages de P_2 :

Page	Cadre	Valide
00	24	0
01	32	0
08	2A	1
09	76	1
0A	54	0
18	6C	1
19	55	1
1A	31	1

10. Quelle est l'adresse physique qui correspondant à l'adresse logique 0B50 pour le processus P_1 ?
11. Quelle est l'adresse physique correspondant à l'adresse logique 0B50 pour le processus P_2 ?
12. Quelle est l'adresse physique correspondant à l'adresse logique 0C0D pour le processus P_1 ?

Exercice 8. Segmentation paginée On considère un système de gestion de mémoire de 64Mo de mémoire physique, gérée de manière segmentée et paginée avec deux niveaux de pagination. Un processus peut avoir au plus 256 segments. Chaque segment peut adresser au plus 64ko de mémoire. Enfin, la taille des cadres de page est fixée à 4ko et le répertoire contient 4 lignes.

1. Quelle est la taille et la composition de l'adresse logique ? Indiquez sa composition.
2. Quelle est la taille et la composition de l'adresse linéaire ?
3. Quelle est la taille et la composition de l'adresse physique ?

Soit un processus muni de la table des segments et du répertoire suivant :

Segments :

Segment	Base	Limite
00	BE 0A 75	05 00
01	BE 23 D1	00 BF
02	BE 00 DA	03 61
03	BE 1A 26	05 D2
04	BE 0F F0	00 CF

Répertoire :

Répertoire	Table	Valide
0	0	1
1	3	0
2	1	1
3	2	0

et de deux tables de pages (on ne fait figurer ici que quelques pages, les autres ont leur bit de validité à 0) :

Table 0 :

Page	Cadre	Valide
2A0	23 40	1
2A1	05 BB	1
2A2	00 00	0
2A3	14 E0	1

Table 1 :

Page	Cadre	Valide
3E0	00 00	0
3E1	27 FD	1
3E2	00 00	0
3E3	3A F6	1

4. Quelle est l'adresse physique correspondant à l'adresse logique 03 00 F0 ? Indiquez clairement les valeurs calculées pendant le processus de traduction.
5. Quelle est la quantité de mémoire physique utilisée par le processus ? Justifiez.

 **Exercice 9. Pagination Sv32 des processeur RISC-V** L'architecture de processeur RISC-V propose différents modes de gestion de la mémoire en fonction des besoins de l'utilisateur. Lorsque les besoins en mémoire sont limités, le mode le plus simple Sv32 est utilisé. Dans ce mode les adresses logiques sont codées sur 32bits, les adresses physiques sur 34bits et une pagination à deux niveaux est utilisée avec des pages standard de 4ko. Le répertoire est les tables de pages contiennent le même nombre de lignes.

1. Quelle est la composition des adresses logiques ?
2. Quelle est la composition des adresses physiques ?

Le répertoire et la table des pages ont la même structure, chaque ligne est composée d'un numéro de cadre sur 22bits et de 10bits d'informations supplémentaires parmi lesquels des bits de protection R, W et X indiquant si la ligne peut être utilisée pour un accès en lecture, écriture ou exécution. Pour le répertoire, ces trois bits sont normalement à zéro.

3. Quelle est la taille du répertoire et d'une table des pages ?

4. Il n'y a que 22bits disponibles dans le répertoire pour indiquer la position en mémoire physique de la table des pages. Pourquoi n'y en a-t-il pas 34 ?

Chaque ligne du répertoire dispose des bits R, W et X qui n'ont pas vraiment de sens si l'on considère qu'une ligne indique seulement où se trouve la table de pages à consulter. Si par contre on considère que le répertoire découpe la mémoire logique en $2^{10} = 1024$ zones de 4Mo qui sont ensuite découpées en $2^{10} = 1024$ pages de 4ko par les tables de pages correspondantes, on peut améliorer la gestion de la mémoire d'une manière intéressante.

Une ligne du répertoire ou les trois bits R, W et X ont pour valeur 0, indiquent que le numéro de cadre pointe vers une table des pages gérant cette zone de 4Mo sous la forme de petites pages de 4ko. Une autre combinaison de ces trois bits indique que cette zone est gérée sous la forme d'une seule grosse page de 4Mo dont l'adresse physique est donnée par le numéro de cadre.

Cette technique permet d'allouer de gros blocs de mémoire contigüe sans avoir besoin d'utiliser de tables de pages ce qui économise de la mémoire.

5. Quelle quantité de mémoire est économisée par l'utilisation d'une "superpage" par rapport à une table de pages normale ?

6. Quel autre avantage présente l'utilisation de ces "superpages" ?

L'architecture RISC-V propose d'autres modes de gestion de la mémoire allant jusqu'à 57 bits d'adresses logiques pour 66 bits d'adresses logiques avec 5 niveaux de pagination. Chaque niveau peut indiquer soit une table de plus bas niveau, soit une superpage de grande taille. Pour ces modes permettant de gérer de très grandes quantités de mémoire, l'utilisation des superpages devient critique pour gérer efficacement la mémoire.