Mémoire virtuelle

INTRODUCTION -1

- Pb : La taille d'un processus doit pouvoir dépasser la taille de la mémoire physique disponible, même si l'on enlève tous les autres processus

- Solution : le SE conserve en mémoire centrale les parties utilisées des processus et stocke, si nécessaire, le reste sur disque →principe de la **mémoire virtuelle**

- La mémoire virtuelle est une technique qui permet d'exécuter des programmes dont la taille excède la taille de la mémoire réelle.

- La mémoire virtuelle fait appel à deux mécanismes : segmentation ou pagination

La pagination -1

- L'espace d'adressage d'un processus est divisé en petites unités *de taille fixe* appelées **pages**

- La MC est elle aussi découpée en unités physiques de *même taille* appelées **cadres**

- taille : défini par le matériel (512octet)
  - variable en fonction du système (512 - 8192 o)

→ 2 Types d'adressage :

- Adressage virtuel

(Numéro de page, déplacement) , (P,dep)

- Adressage réel ou physique→(Numéro de case, déplacement), (C,dep)

La pagination -2

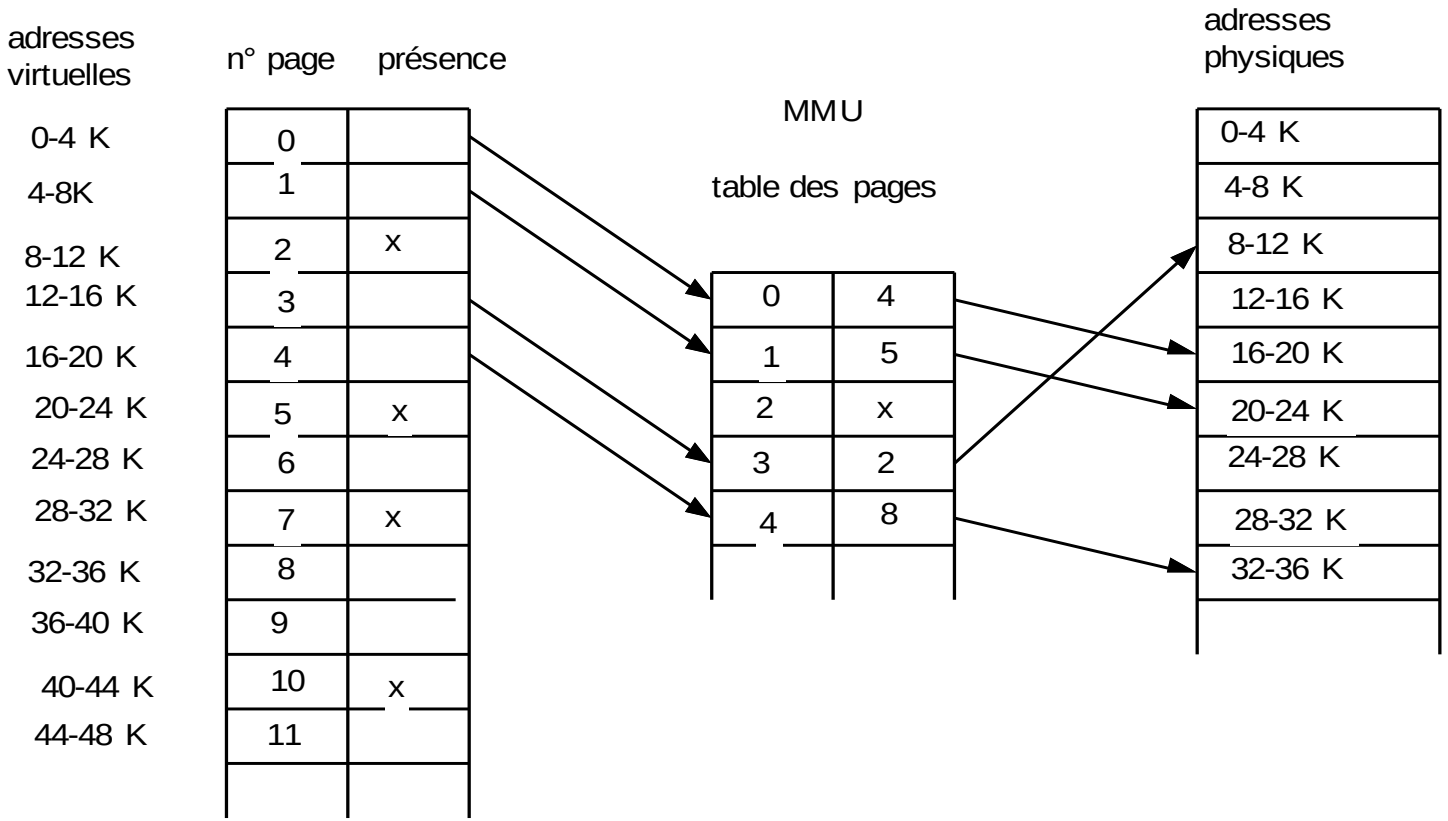
- Pour un processus, le système ne chargera que les pages utilisées
- Mais la demande de pages à charger peut être plus élevée que le nombre de cadres disponibles
- Une gestion de l'allocation des cadres libres est nécessaire.
- Dans un SE à pagination, **la MMU** (Memory Management Unit ou unité de gestion de la mémoire) traduit les adresses virtuelles en adresses physiques.

La pagination -3

Correspondance entre adressages (virtuel/réel)

- La correspondance entre les pages et les cases est mémorisée dans une table appelée Table de pages (TP).
- Chaque entrée de la Table de pages est composée de plusieurs champs, notamment :
  - 1. Le bit de présence.
  - Le bit de modification (M).
  - Le numéro de case correspondant à la page.

- Une page est **mappée** ou **chargée** si elle est physiquement présente en mémoire.



- Pages de taille de 4 Ko
- L'adresse virtuelle 12292 correspond à un déplacement de 4 octets dans la page virtuelle 3 (car 12292 = 12288 + 4 et 12288 = 12*1024).
- La page virtuelle 3 correspond à la page physique 2.
- L'adresse physique correspond donc à un déplacement de 4 octets dans la page physique 2, soit : (8*1024) + 4 = 8196 .

La pagination -5

- la page virtuelle 2 n'est pas mappée. Une adresse virtuelle comprise entre 8k et 12k donnera lieu à un défaut de page
- Il y a défaut de page quand il y a un accès à une adresse virtuelle correspondant à une page non mappée.
- Un défaut de page provoque un déroutement (ou TRAP) dont le rôle est de ramener à partir du disque la page manquante référencée.

Algorithmes de remplacement de page-1

- A la suite d'un défaut de page, le système d'exploitation doit ramener en mémoire la page manquante à partir du disque
- S'il n'y a pas de cadres libres en mémoire, il doit retirer une page de la mémoire pour la remplacer par celle demandée
- Si la page à retirer a été modifiée depuis son chargement en mémoire, il faut la réécrire sur le disque.

→Quelle est la page à retirer de manière à minimiser le nombre de défauts de page

Algorithmes de remplacement de page-2

- Des algorithmes de remplacement de page ont été proposés
- Il y a plusieurs algorithmes de remplacement de page :
  - l'algorithme de remplacement aléatoire qui choisit au hasard la page victime
  - l'algorithme optimal de Belady,
  - l'algorithme de remplacement de la page non récemment utilisée (NRU),
  - l'algorithme FIFO,
  - l'algorithme de la page la moins récemment utilisée,
  - et bien

Remplacement de page optimal (Algorithme de Belady) -1

- consiste à retirer la page qui sera référencée le plus tard possible dans le futur
- Cette stratégie est impossible à mettre en œuvre
  - il est difficile de prévoir les références futures d'un programme
- cependant elle a été utilisé comme base de référence pour les autres stratégies
  - elle minimise le nombre de défauts de page

Remplacement de page optimal (Algorithme de Belady) -2

- **Exemple .** Soit un système avec m= 3 cases de mémoire et une suite de références ={7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1

- Les entrées en **noir** sont des pages chargées après un défaut de page
- L'algorithme optimale fait donc **9 défauts** de page

m \ ω	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
0	**7**	7	7	**2**	2	2	2	2	2	2	2	2	2	2	2	2	2	**7**	7	7
1		**0**	0	0	0	0	0	**4**	4	4	**0**	0	0	0	0	0	0	0	0	0
2			**1**	1	1	**3**	3	3	3	3	3	3	3	**1**	1	1	1	1	1	1

Remplacement de la page non récemment utilisée (NRU, Not Recently Used) -1

- Cet algorithme utilise les bits **R et M** associés à chaque page pour déterminer les pages non récemment utilisées.
  - Le bit **R est positionné à 1 chaque** fois qu'une page est référencée
  - Le bit **M est positionné lorsque la page est modifiée** (elle n'est plus identique à sa copie sur disque)

Remplacement de la page non récemment utilisée (NRU, Not Recently Used) -2

- Lorsqu'un défaut de page se produit, l'algorithme NRU sélectionne la page à retirer en procédant comme suit :

  - Il vérifie s'il existe des pages non référencées et non modifiées (**R=0** et **M=0).** Si c'est le cas, il sélectionne une page et la retire.

  - Sinon, il vérifie s'il existe des pages non référencées et modifiées (**R=0** et **M=1).** Si c'est le cas, il sélectionne une page et la retire (une sauvegarde sur disque de la page retirée est nécessaire).

Remplacement de la page non récemment utilisée (NRU, Not Recently Used) -3

- Sinon, il vérifie s'il existe des pages référencées et non modifiées (**R=1** et **M=0).** Si c'est le cas, il sélectionne une page et la retire.

- Sinon, il sélectionne une page référencée et modifiée et la retire (**R=1** et **M=1).** Dans ce cas, une sauvegarde sur disque de la page retirée est nécessaire.

Remplacement de page FIFO -1

- Il mémorise dans une file de discipline FIFO (premier entré, premier sorti) les pages présentes en mémoire.

- Lorsqu'un défaut de page se produit, il retire la plus ancienne, c'est à dire celle qui se trouve en tête de la file.

REMPLACEMENT DE PAGE FIFO -2

- **Exemple** La suite de référence w={7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1} avec m=cases fait 15 défauts de page avec l'algorithme FIFO

m \ w	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
0	**7**	7	7	**2**	2	2	2	**4**	4	4	**0**	0	0	0	0	0	0	**7**	7	7
1		**0**	0	0	0	**3**	3	3	**2**	2	2	2	2	**1**	1	1	1	1	**0**	0
2			**1**	1	1	1	**0**	0	0	**3**	3	3	3	3	**2**	2	2	2	2	1

Remplacement de la page la moins récemment utilisée (LRU Least Recently Used)

- L'algorithme LRU mémorise dans une liste chaînée toutes les pages en mémoire. La page la plus utilisée est en tête de liste et la moins utilisée est en queue.

- Lorsqu'un défaut de page se produit, la page la moins utilisée est retirée

Remplacement de la page la moins récemment utilisée (LRU Least Recently Used) -2

- **Exemple :** La suite de références={7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1}

  avec m=3 cases fait 12 défauts de page avec l'algorithme de remplacement de la page la moins récemment utilisée

m \ ω	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
0	**7**	7	7	**2**	2	2	2	**4**	4	4	**0**	0	0	1	1	1	1	**1**	1	1
1		**0**	0	0	0	0	0	**0**	0	**3**	3	3	3	3	3	**0**	0	0	0	0
2			**1**	1	1	**3**	3	3	**2**	2	2	2	2	2	2	2	2	**7**	7	7