

Gestion de la mémoire par pages

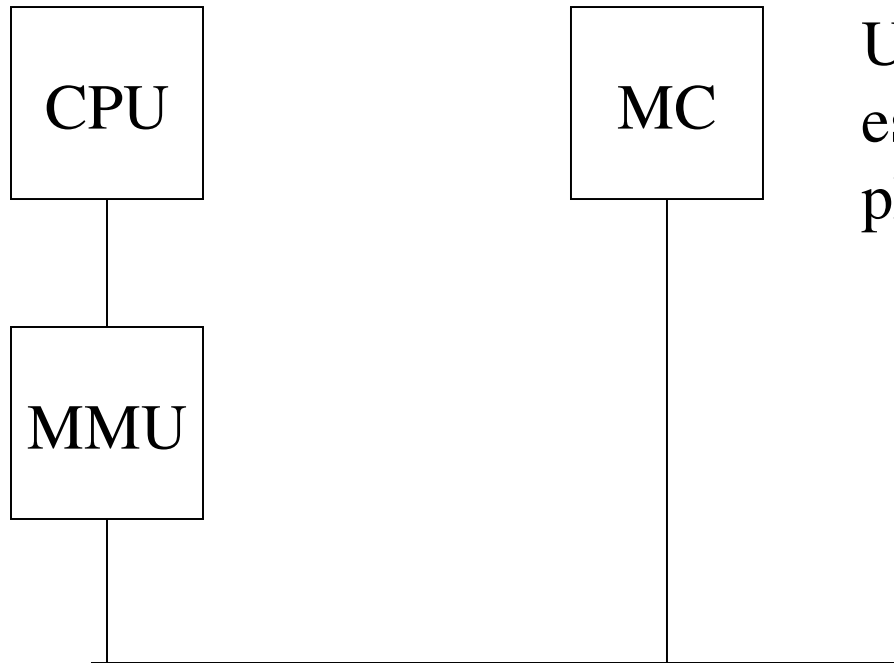
Plan du chapitre

- Motivation
- Traduction dynamique d'adresses par pagination
- Généralités sur la mise en œuvre
- Comportement des programmes
- Remplacement de pages
- Gestion globale de la mémoire

Motivation

- Indépendance des tailles des processus et de la mémoire principale
- On ne veut charger en mémoire centrale que les zones utiles
- Solution : utiliser un espace virtuel découpé en blocs pouvant être placés en mémoire indépendamment les uns des autres
- Rappel : la mémoire virtuelle fait partie du processeur

La mémoire virtuelle

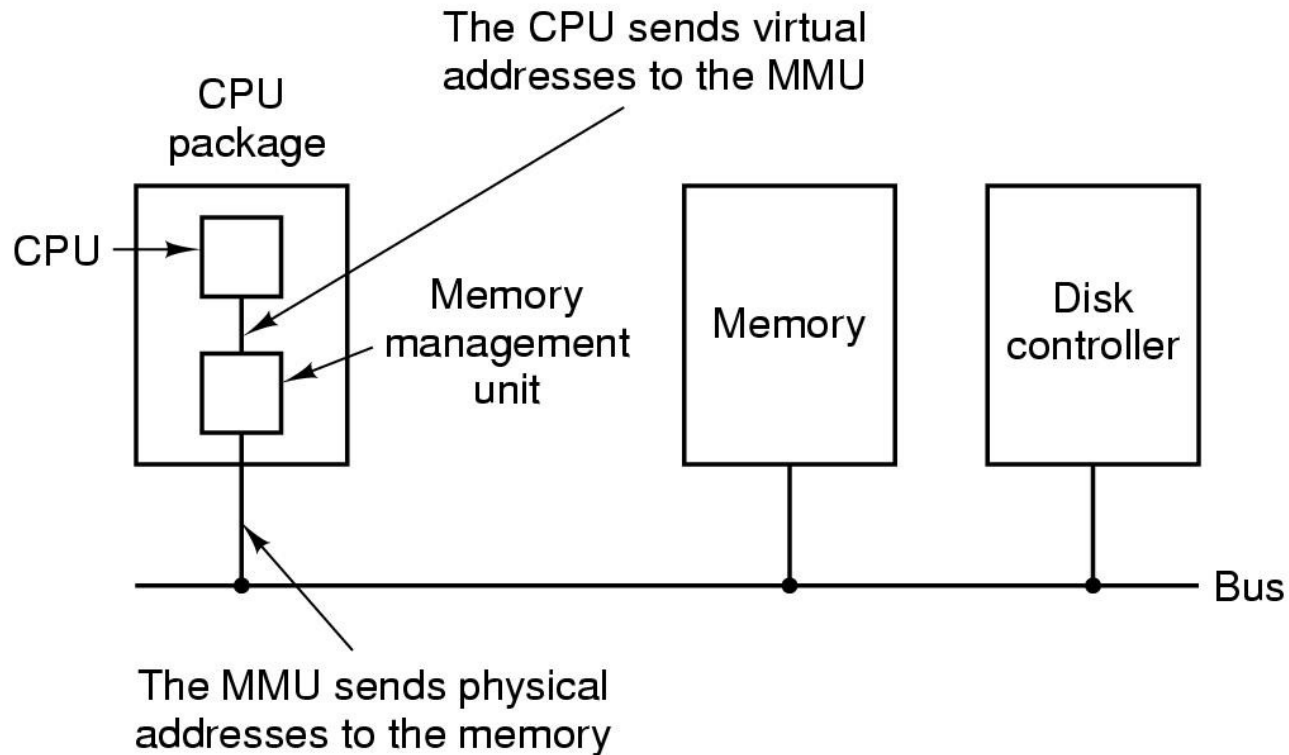


Une adresse virtuelle
est transformée en adresse
physique par le MMU

Traduction dynamique d'adresses par pagination

- Mémoire virtuelle divisée en zones de taille fixe appelées **pages**
 - Taille typique d'une page: 4 Koctets
 - Taille mémoire virtuelle : 32 ou 64 bits d'adresse
- Mémoire physique divisée en blocs de même taille que les pages appelés les **cases**
- Une page peut être chargée dans n'importe quelle case

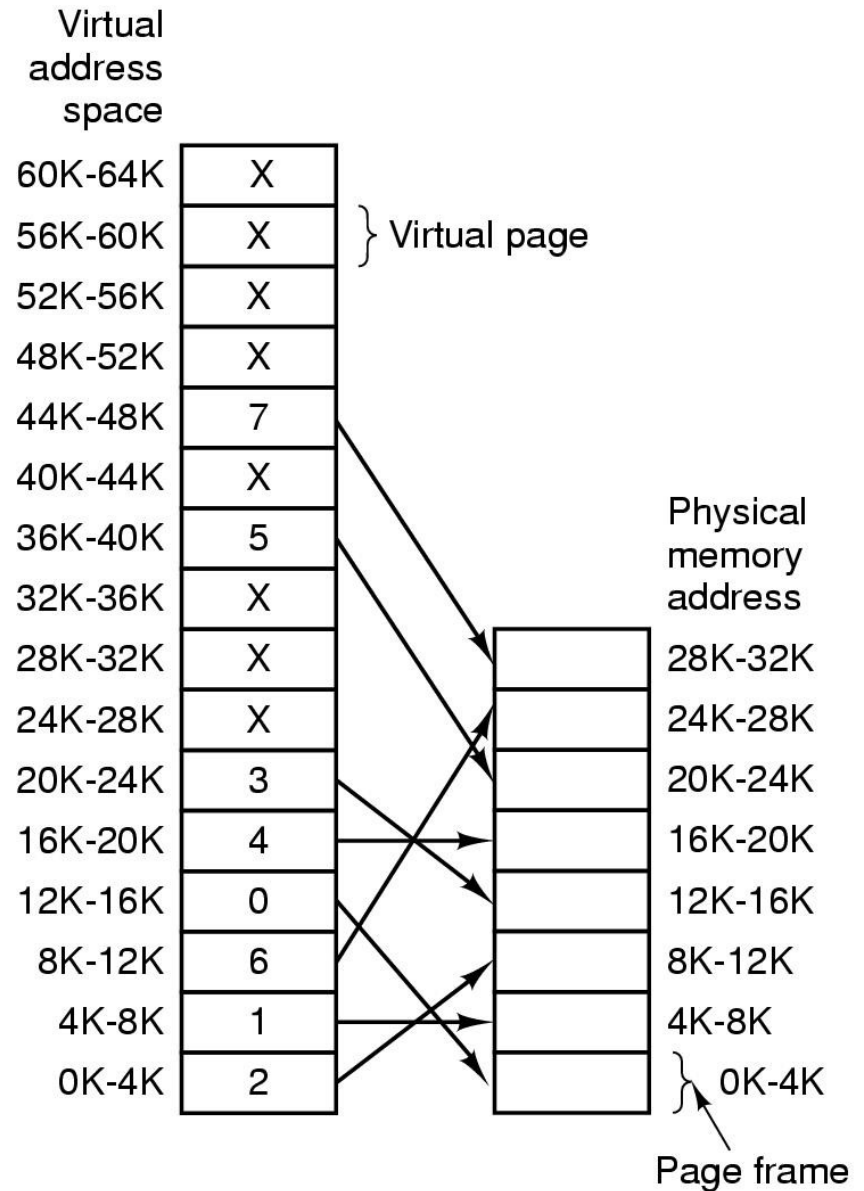
Architecture globale



The position and function of the MMU

Pagination

The relation between
virtual addresses
and physical
memory addresses
given by
page table



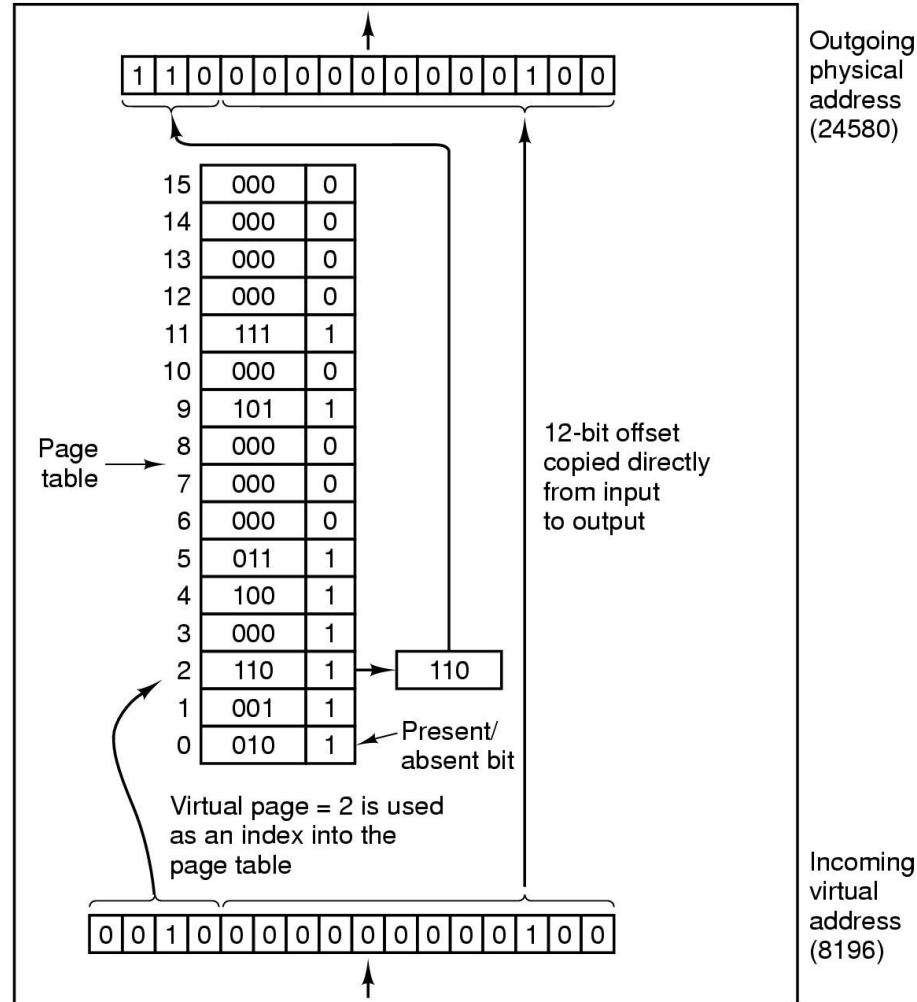
Que se passe-t-il si la traduction échoue ?

- Accès à une page non chargée en mémoire
- Le MMU ne peut donc pas calculer l'adresse physique correspondant à l'adresse virtuelle entrée
- Il y aura donc un déroutement appelé le déroutement pour défaut de page (page fault)
 - Le traitement de ce déroutement déclenche le chargement de la page absente

Comment le MMU fait il la traduction ?

- Problème économique : la mémoire centrale est de plus en plus grande
- Registres intégrés (mémoire topographique)
- Table de pages en mémoire centrale
- Efficacité : 2 accès sont nécessaires au lieu d 'un seul
- Table de pages + registres associatifs

Page Tables (1)

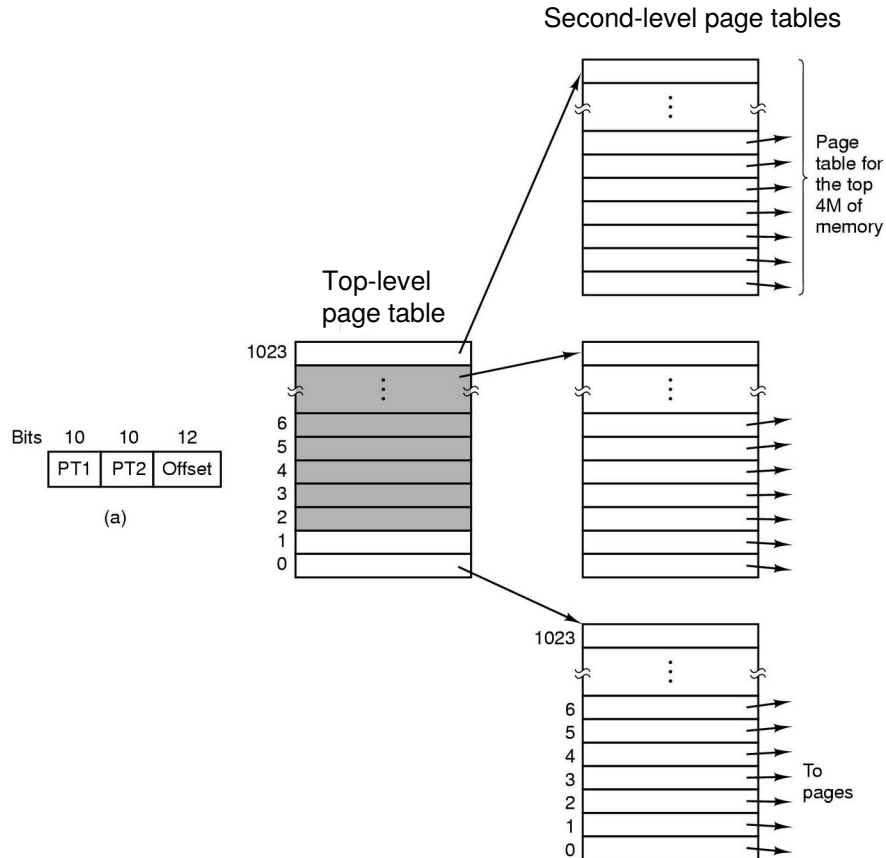


Internal operation of MMU with 16 4 KB pages

Compléments

- Tables de pages structurées en arbre
 - Pagination à 2, 3 ou 4 niveaux
- Pagination sans table de pages
 - Seuls sont présents des registres associatifs
 - On peut alors introduire de très grandes mémoires virtuelles (64 bits)
 - L 'organisation des tables de page est laissée au système

Page Tables (2)

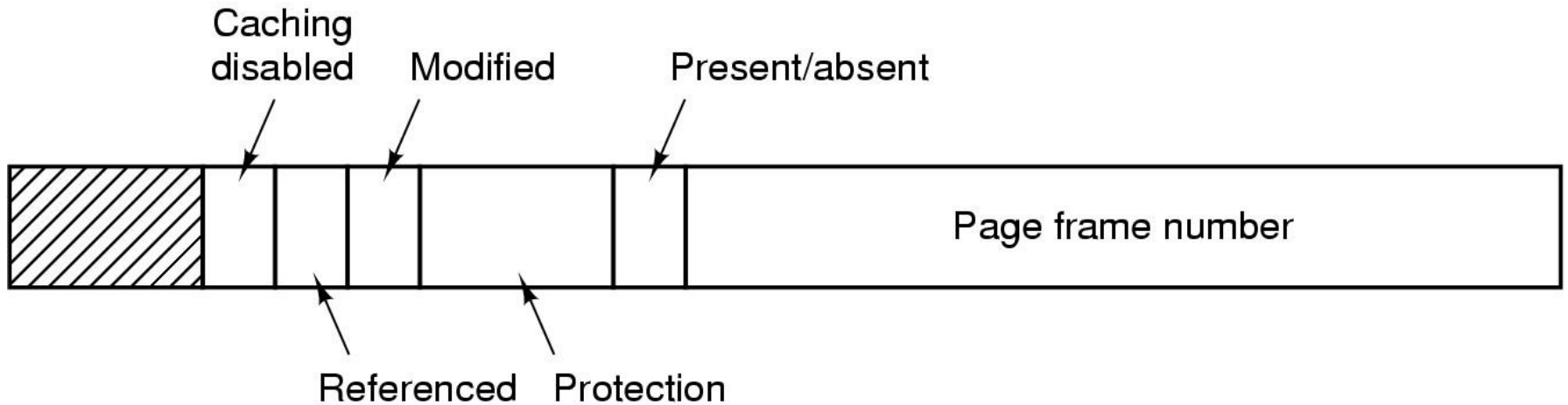


- 32 bit address with 2 page table fields
- Two-level page tables

Quelles informations associer à chaque page

- Le numéro de la case dans laquelle la page est chargée
- Un bit P de présence en mémoire physique
- Des informations sur les droits d'accès
 - Lecture, écriture, exécution
- Un bit indiquant si la page a été modifiée depuis son chargement en mémoire

Page Tables (3)



Typical page table entry

TLBs – Translation Lookaside Buffers

Valid	Virtual page	Modified	Protection	Page frame
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

A TLB to speed up paging

Généralités sur la mise en œuvre

- Un espace virtuel est associé à chaque processus
- Quand un processus s'exécute :
 - Si défaut de page, alors allocation d'une case et transfert de la page dans la case allouée
- Si la mémoire est pleine, un algorithme de **remplacement** choisit une case dont on vide le contenu sur disque
- C'est efficace si les défauts de page sont rares

Comportement des programmes

- Exécution d 'un programme = suite de références aux pages
- Distribution des références sur toute une exécution
- Comportement en mémoire restreinte
- Comment « prévoir » les pages qui seront utilisés dans le futur proche

Distribution des références

- Pas de distribution uniforme
- L 'histogramme fait apparaître des pics marqués
- Justification
 - Code mort
 - Traitement des erreurs
- 25% des pages, 75 % des références

Comportement en mémoire restreinte

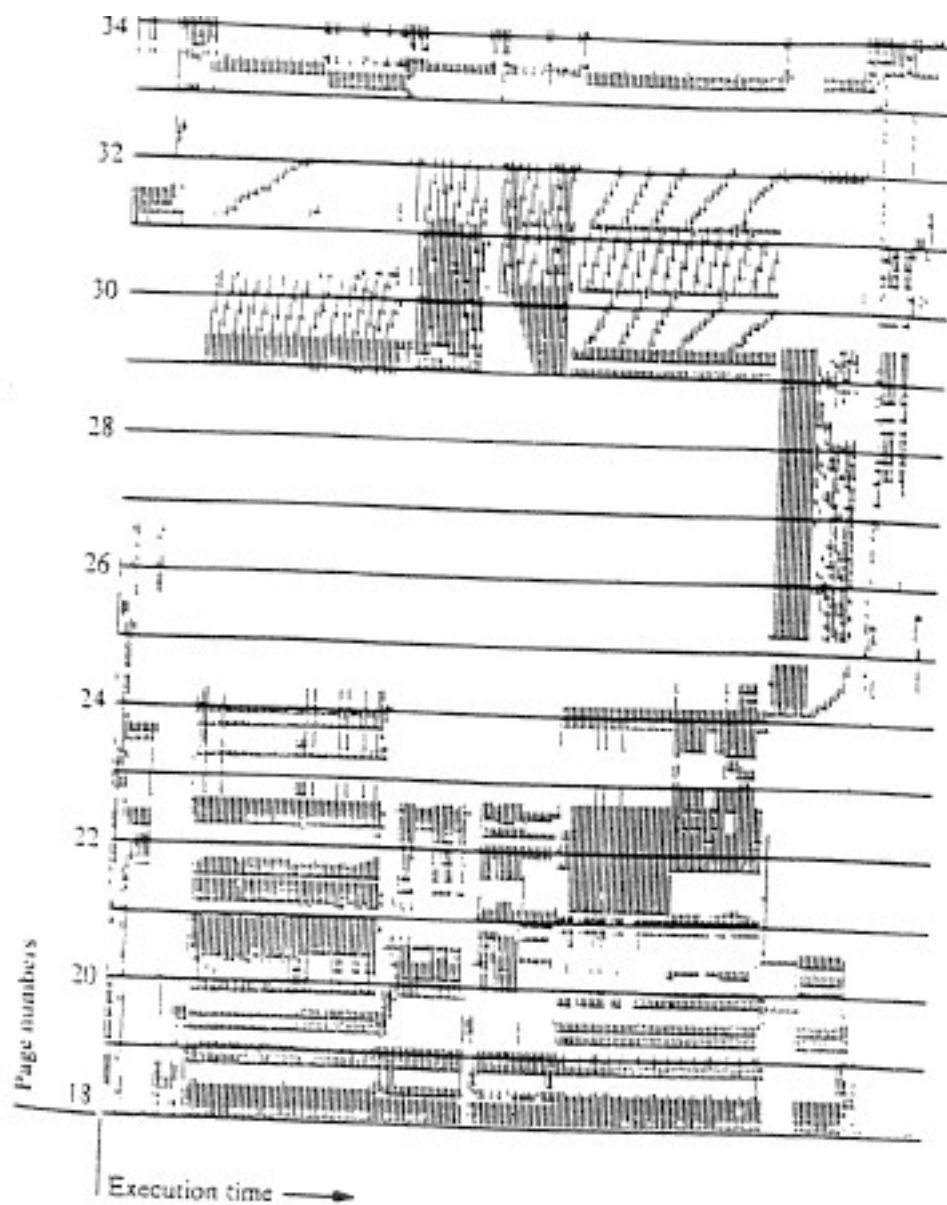
- Un processus s'exécute sur n cases
 - (n petit par rapport à la taille du programme)
- En cas de défaut de page, un algorithme de remplacement détermine la case à vider
- On mesure soit le nombre total de défauts de page, soit l'intervalle entre deux défauts de page
- On observe une taille de mémoire critique

Rappel du cours et TD

- Mémoire virtuelle, pagination
- Comportement des programmes
- En TD, premier exemple de mise en œuvre
 - Gestion des cases libres
 - Traitement des défauts de page

Propriété de localité

- Stabilité des références sur une brève période
- Les références observées sur un passé récent sont en général une bonne estimation des prochaines références



Ensemble de travail

- « working set »
- $W(t,T)$ = ensemble des pages référencées entre $t-T$ et t
- $W(t,T)$ est une bonne approximation de $W(t+T,T)$

Algorithmes de remplacement(1)

- Processus unique s'exécutant sur un nombre fixe de cases
- Algorithme optimal
- FIFO
 - Remplacement dans l'ordre des chargements
- RAND
 - Remplacement d'une case choisie au hasard

Algorithmes de remplacement (2)

- Algorithme de la seconde chance
 - On doit disposer d 'un bit d 'utilisation
- Algorithme de l 'horloge
- Algorithme LRU
 - Remplacement de la case la moins récemment utilisée

Comparaison des algorithmes de remplacement

- RAND et FIFO sont équivalents
 - RAND n 'est donc jamais utilisé
- LRU donne les meilleurs résultats
 - Mais il est difficile à mettre en œuvre en l 'absence d 'aide matérielle
- On essaie donc d 'approcher LRU
 - L 'horloge et la seconde chance appartiennent à cette classe

Remplacement local ou global

- Cas de plusieurs processus
- Remplacement global
 - L 'algorithme de remplacement est exécuté sur l 'ensemble des cases de mémoire
 - On peut alors voler une case à un processus en attente de page
- Remplacement local
 - L 'algorithme de remplacement ne concerne que les cases affectées au processus courant

Architecture globale d'un système paginé

- Un exemple simpliste
 - États des processus
 - Structures de données
- Gestion globale de la mémoire

Un exemple simpliste (1)

- Un espace virtuel par processus (une table de pages)
- Les pages des processus bloqués sont conservées sur disque
- Chargement des pages à la demande
- Lorsqu'un processus attend une page, il est placé dans un état dit Suspendu
- Un algorithme de remplacement détermine quelle case vider

Un exemple simpliste (2)

- Diagramme d'états des processus
- Actions effectuées lors des transitions
- Structures de données associées aux pages
 - Table des pages avec adresse disque et numéro de case
- Structures de données associées aux cases
 - Table des cases

Un exemple simpliste (3)

- Fonctionnement convenable si le système est peu chargé
- On fait alors varier le nombre des processus
- Observation : le phénomène d'écroulement (« thrashing »)
- Explications simples
- Toute bonne gestion de la pagination doit éviter l'écroulement

Gestion globale de la mémoire

- Éviter l'écroulement en réduisant le nombre de processus admis en mémoire
 - Nouvel état pour les processus : prêt
- Méthode de régulation
- Espace de travail

Gestion globale par régulation

- Principe
- Système surchargé, sous-charge, normal
- Si surcharge, on chasse un processus de la mémoire
- Si sous-charge, on admet en mémoire un processus prêt
- Taux d 'utilisation du disque, activité de l 'UC

Gestion fondée sur l 'espace de travail

- Principe : tenir compte du comportement individuel de chaque processus
- On n 'admet en mémoire un processus prêt que si on peut lui réserver un nombre de cases égal à la taille de son espace de travail
- Remplacement local
- Peut s 'accompagner ou non de préchargement

Compléments

- Entrées-sorties et espace virtuel
- Initialisation d'un espace virtuel
- Accès aux fichiers par couplage
- Partage d'informations entre espaces virtuels

Entrées-sorties et espace virtuel

- Rappel sur les entrées-sorties, DMA
- Un DMA assure le transfert entre un périphérique et une zone utilisateur
 - DMA utilise des adresses physiques
- Que faire si la zone utilisateur est absente ou vidée au cours d'une E/S ?
- Solution 1: page collée en mémoire
- Solution 2 : tampons du système
 - Comment transférer entre zones utilisateur et système

Définition de l'espace virtuel d'un processus

- Rappel : pour chaque espace virtuel, des tables décrivent la localisation des pages sur disque et ou en mémoire.
- Quand ces tables sont elles initialisées ?
 - Création d'un processus
- Espace disque pour le va et vient
- Initialisation d'un processus en mémoire
 - Allocation de l'espace disque au remplacement

Accès aux fichiers par couplage (1)

- Un transfert de page = E/S disque
- Un accès au fichier = E/S disque
- Idée : uniformiser les mécanismes en mettant en correspondance un fichier ou une partie de fichier avec une plage d'espace virtuel
- Inconvénient : copie implicite par algorithme de remplacement.
- Avantage : mécanisme unificateur

Accès aux fichiers par couplage (2)

- Coupler (fichier, adr. virtuelle, depl.fichier, longueur)
- Comment accéder à un fichier par couplage ?
- Comment recopier un fichier dans un autre ?
- Retour sur la création d 'un processus

Partage entre espaces virtuels

- Principe simple : 2 processus font des couplages à un même fichier
- Questions :
 - À quelles adresses virtuelles ?
 - Combien de pages en mémoire centrale ?
 - Plusieurs : cohérence
 - Une ?
 - Technique : adapter l'algorithme de remplacement

Problèmes d'adressage liés au partage

- Partage de fichier de données
 - Absolues
 - Contenant des pointeurs
- Partage de procédures
- Partage de programmes complets
 - Code
 - Données globales

Récapitulation (1)

- Espace virtuel
 - Découplage entre les adresses vues du programmeur et les adresses de chargement
 - Un programme peut être chargé n'importe où en mémoire
- Pagination
 - Un programme peut être chargé dans des emplacements non contigus
 - Plus de fragmentation : elle est cachée dans les pages

Récapitulation (2)

- Défaut de page : accès à une page non présente en mémoire
- Défaut de TLB : la MMU ne peut traduire une adresse virtuelle
- Remplacement
 - Case libre choisie en priorité
 - Sinon libération d'une case avec recopie éventuelle de son contenu

Récapitulation (3)

- Gestion globale de la mémoire
 - Maintenir un pourcentage de cases libres
 - Limiter le nombre de processus présents en mémoire
 - Régulation
 - Working set