

Architecture des ordinateurs

Gestion de la mémoire centrale

Philippe Darche
IUT Paris Descartes



Rappel

- A l'origine
 - une petite mémoire primaire (quelques Ko)
 - pour un seul programme
 - pour un seul utilisateur

Le problème

- Contexte de la multiprogrammation
 - plusieurs programmes en cours d'exécution sur un seul processeur
 - plusieurs utilisateurs connectés
- Risques
 - utilisation involontaire ou volontaire d'une ressource
 - ⇒ besoin d'une protection
 - au départ logicielle (prototypage)
 - aujourd'hui matérielle (performance)
- Besoin d'un grand espace mémoire
 - taille limitée de la mémoire physique
 - ⇒ la réponse : le mécanisme de mémoire virtuelle

Gestion de la mémoire primaire

- Le cours ne concerne que la mémoire vive
 - le BIOS (ROM) n'est pas concerné
- Les mécanismes étudiés
 - segmentation
 - pagination

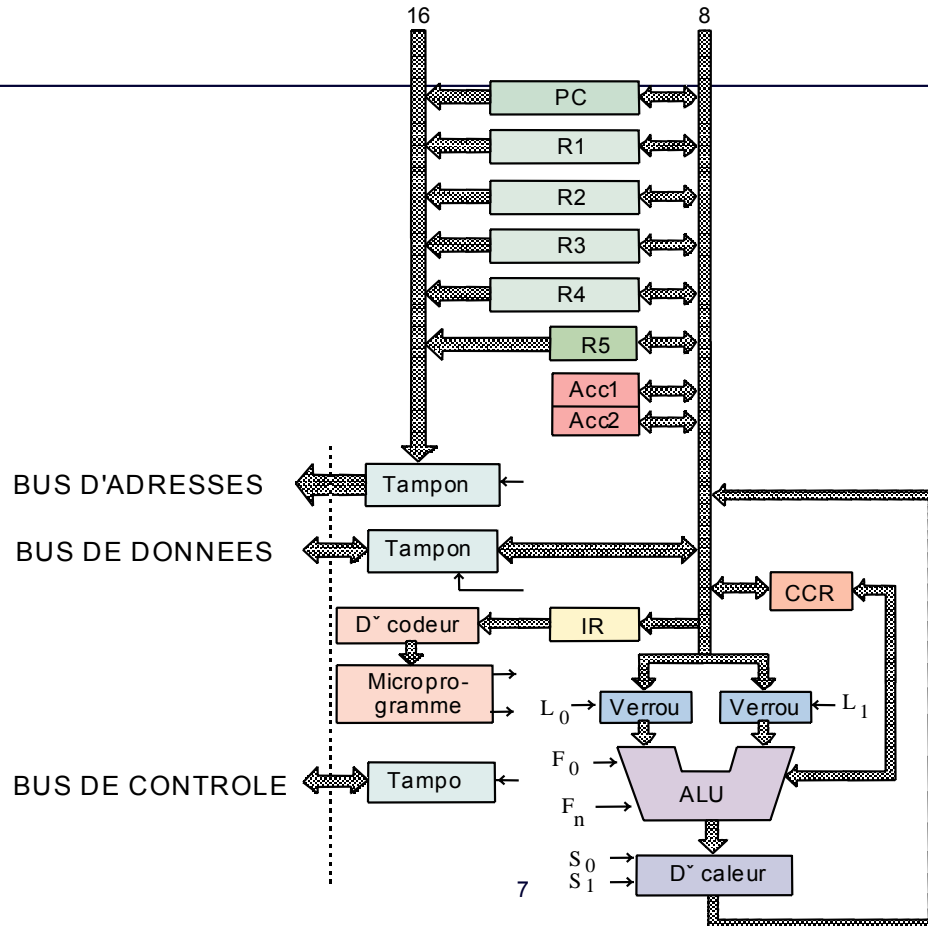
Rappel : L'architecture Von Neumann

- Une abstraction (*i.e.* un modèle théorique)
- Pour juste donner les détails nécessaires à la programmation
- A ne pas confondre avec :
 - l'organisation
 - le modèle de calcul

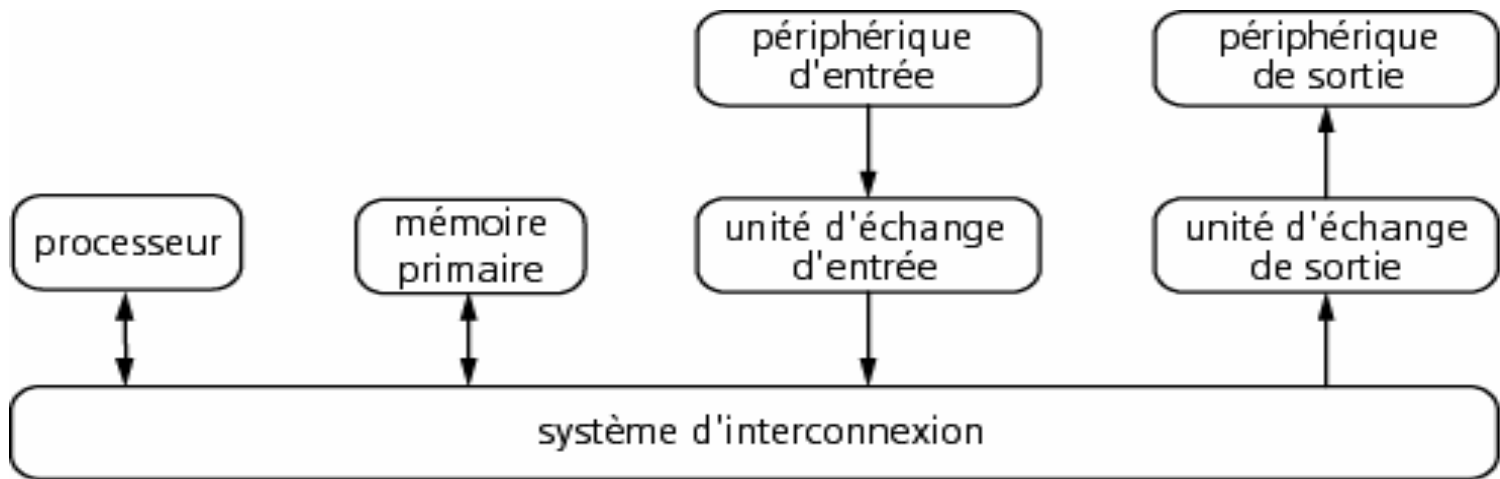


Rappel : L'architecture Von Neumann

Rappel : L'architecture Von Neumann

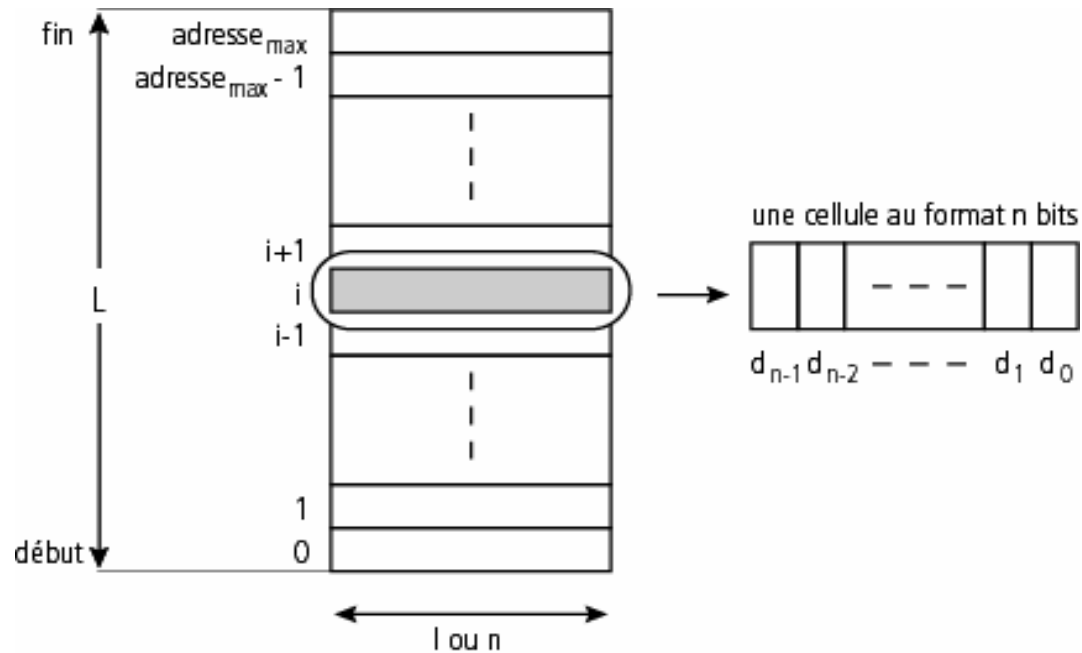


L'organisation d'un ordinateur

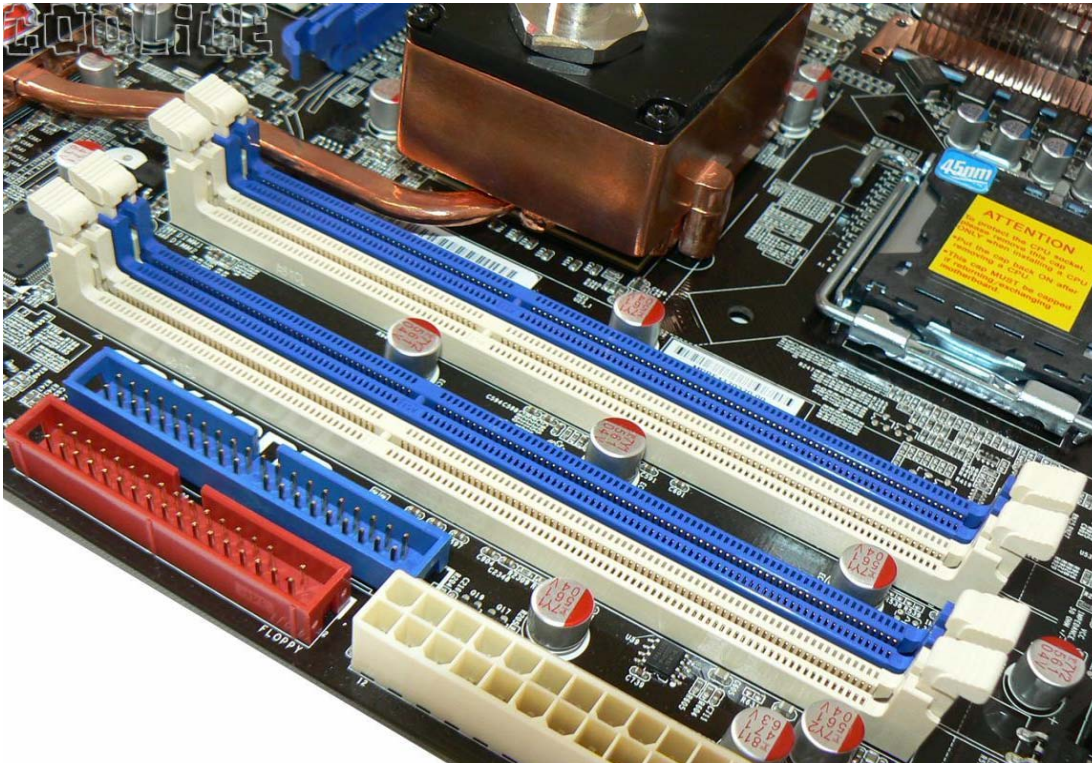


Une mémoire centrale

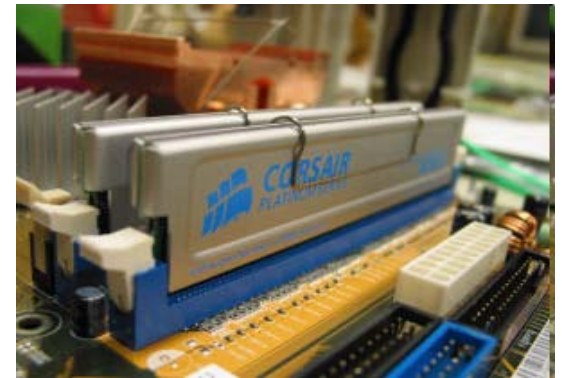
- Taille totale = $L \times l$



Dans un PC



Philippe Darche



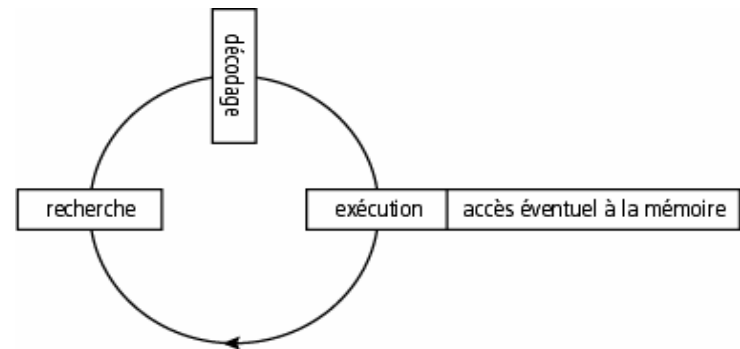


Rappel

- Le programme (suite d'instructions et de données)
 - mémorisé sur un support de mémorisation secondaire
 - est chargé en mémoire centrale pour être exécuté ou interprété

Cycle d'exécution d'une instruction

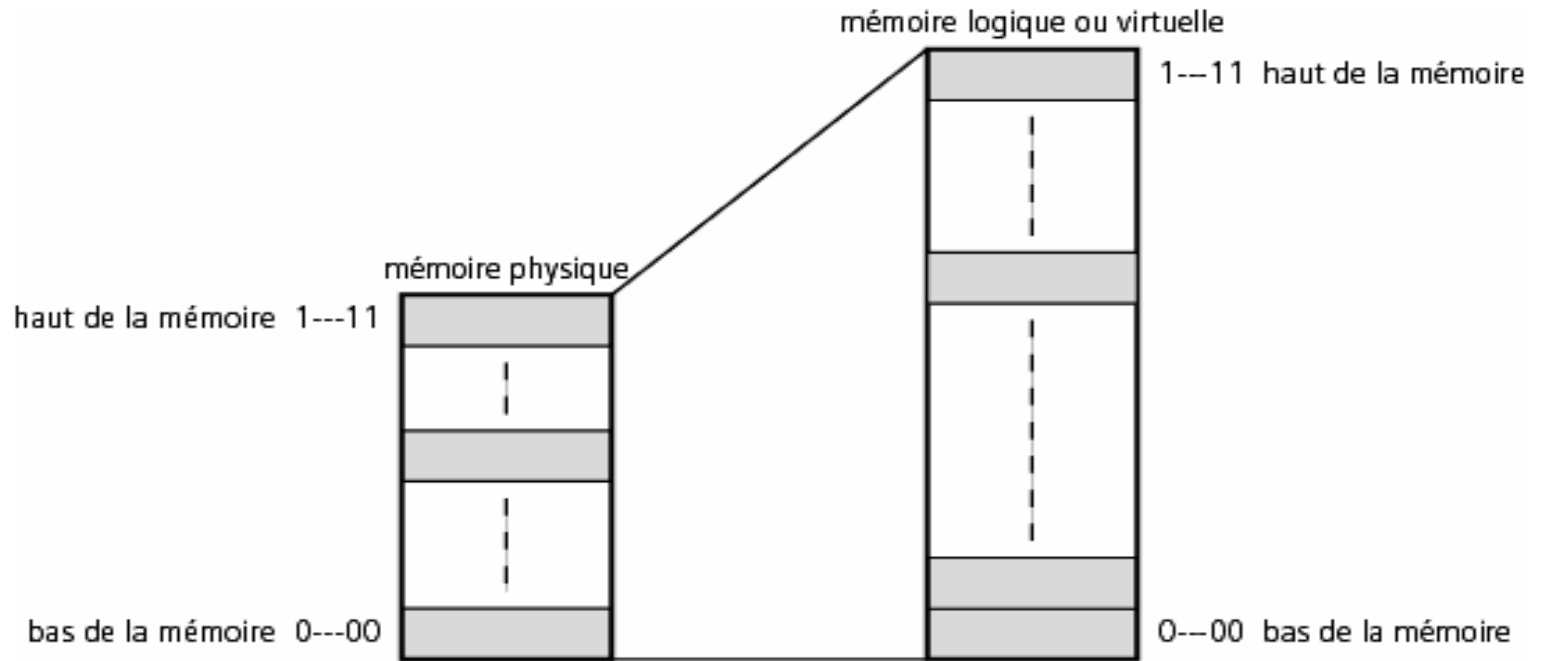
- Un mouvement perpétuel !
- Trois étapes de base
 - recherche (*fetch*)
 - décodage (*decode*)
 - exécution (*execute*)



La mémoire virtuelle

- Zone mémoire logique de taille très supérieure à la mémoire centrale
 - ordre de grandeur du rapport : 2^{10}
- Repose sur l'utilisation d'une mémoire secondaire
 - l'unité de mémoire de masse à disques durs
- Taille limite : celle de la mémoire de masse
- Synonyme : mémoire logique

La mémoire virtuelle



Ordre de grandeur

□ i286 :

- 16 Mo de mémoire physique
- 1 Go de mémoire virtuelle
= 16 384 segments de 64 Ko.

□ i386 :

- 4 Go de mémoire physique ($m = 32$ bits)
- 64 To de mémoire virtuelle
= 16 384 segments de 4 Go



Ordre de grandeur (suite)

- Pentium Pro

- 64 Go de mémoire physique ($m = 36$ bits)
- 64 To de mémoire virtuelle

La compilation

- Traduction identificateur en adresse (logique)
 - variable
 - fonction (adresse de début)

- Exemple

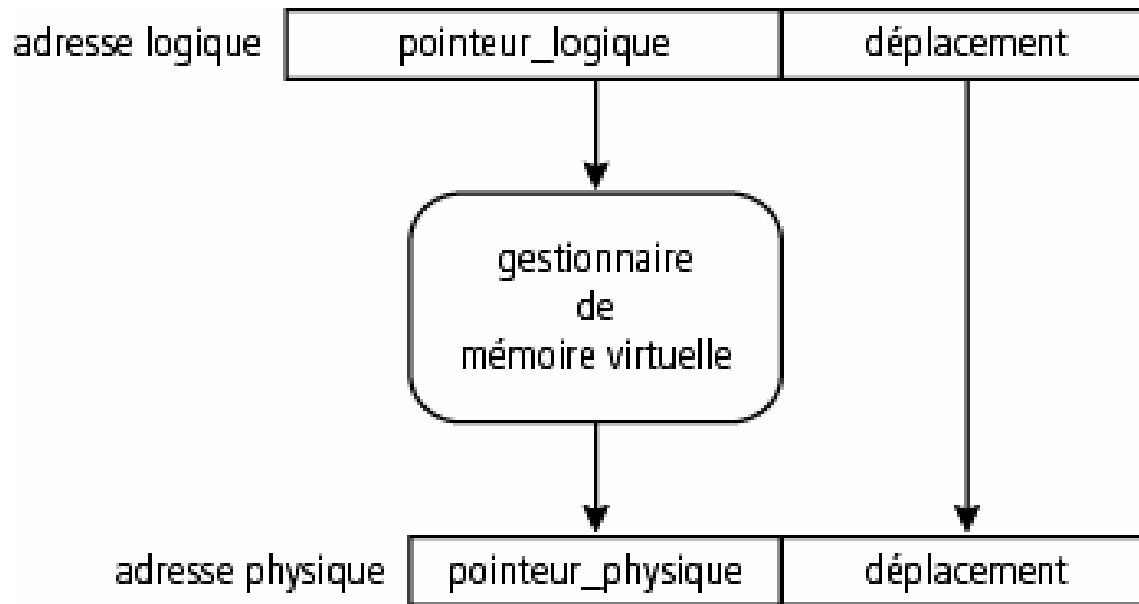
$A = A + 1$ (langage C)

équivalent en langage assembleur

```
mov AX,[A]
add AX,1
mov [A], AX
```

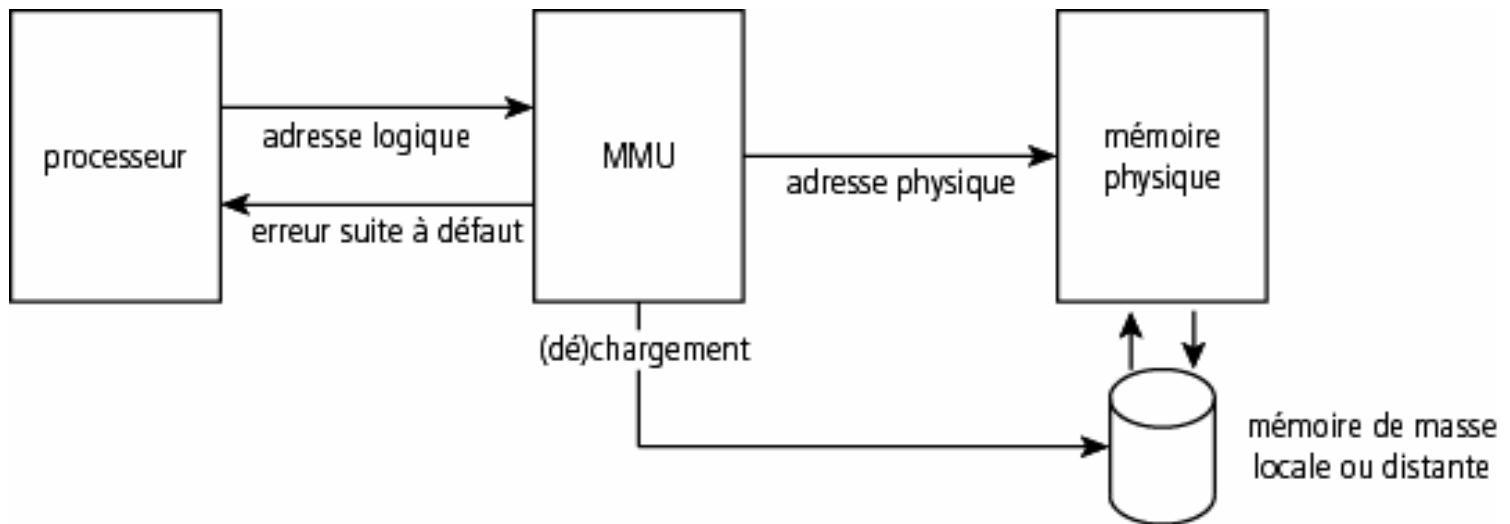
Transformation

- MMU : *Memory Management Unit*

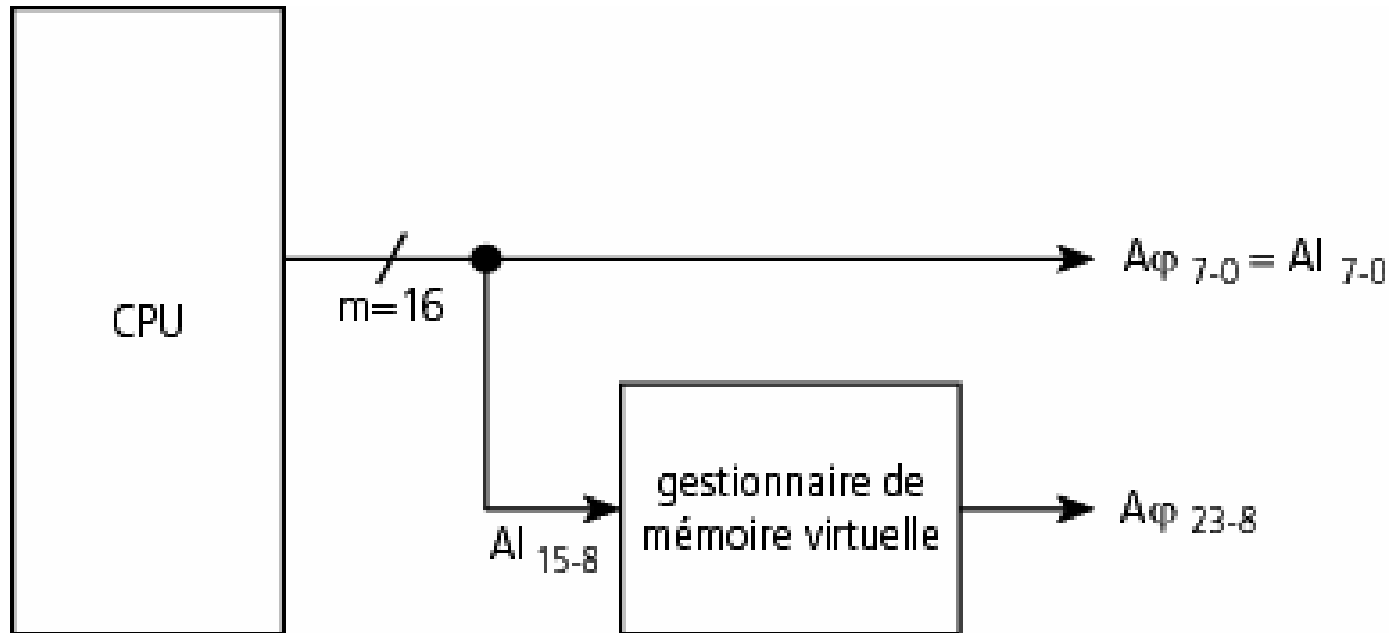


Transformation

- Adresse logique → adresse physique



Un chemin d'adresse

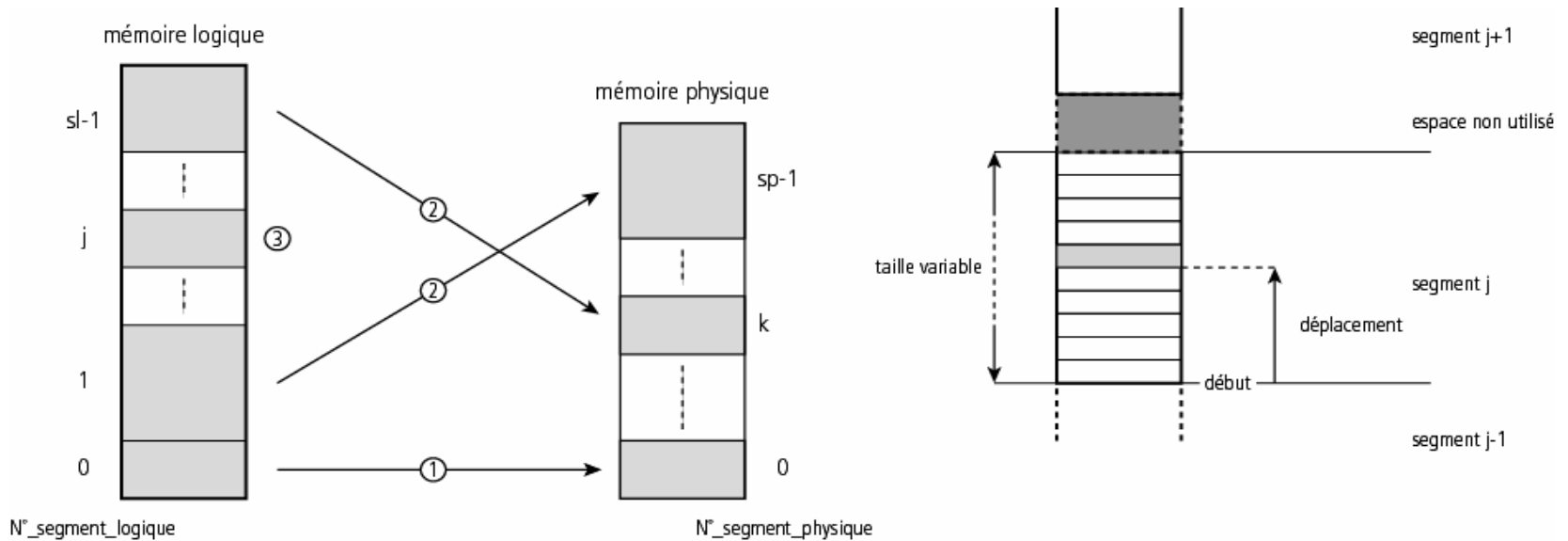


La segmentation

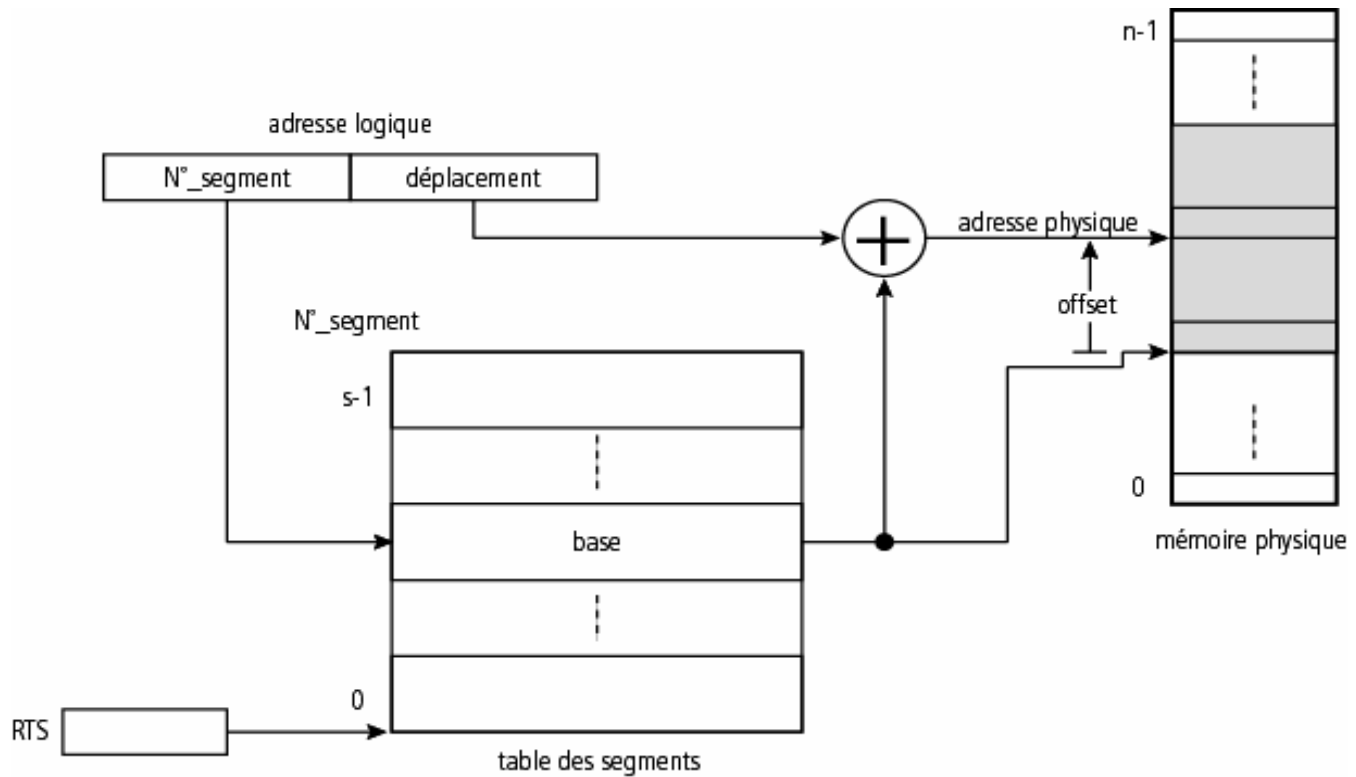
- Un segment
 - zone contigüe de mémoire
 - taille variable avec un maximum
 - i386 → 64 Ko
 - repéré en mémoire par `adr_début_seg`
 - chaque mot est adressable par rapport au début du segment (*offset* ou décalage)
 - adressage relatif

Les segments

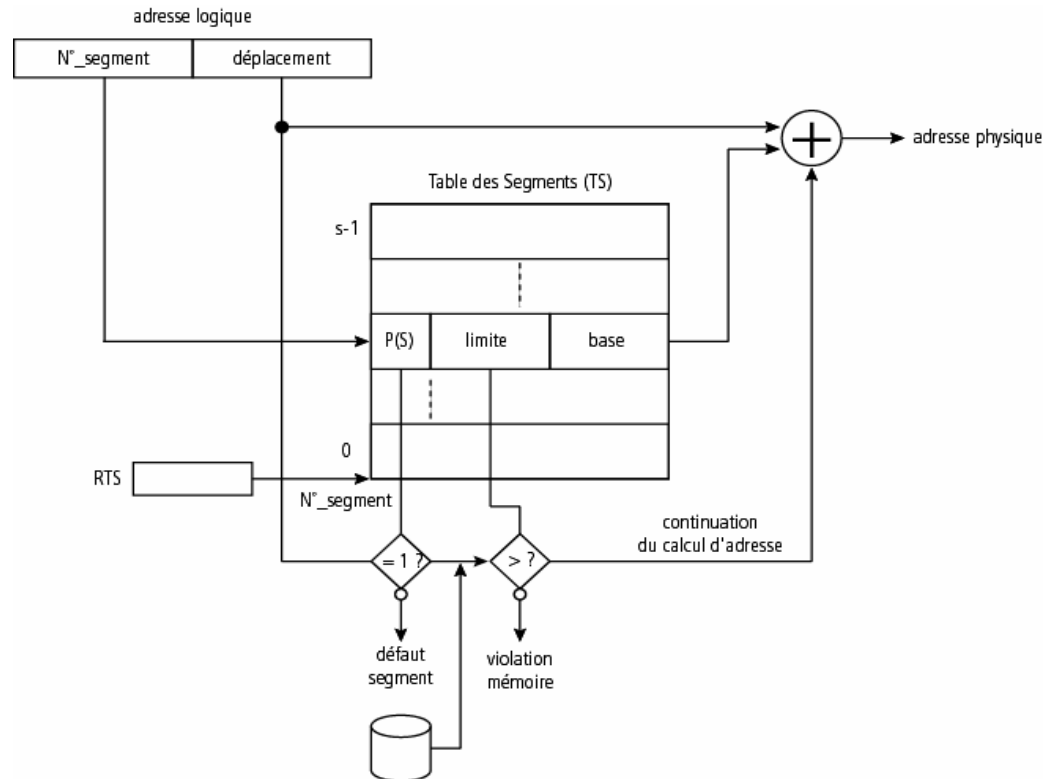
□ Segment en gris clair



Calcul de l'adresse physique



Mécanismes de protection

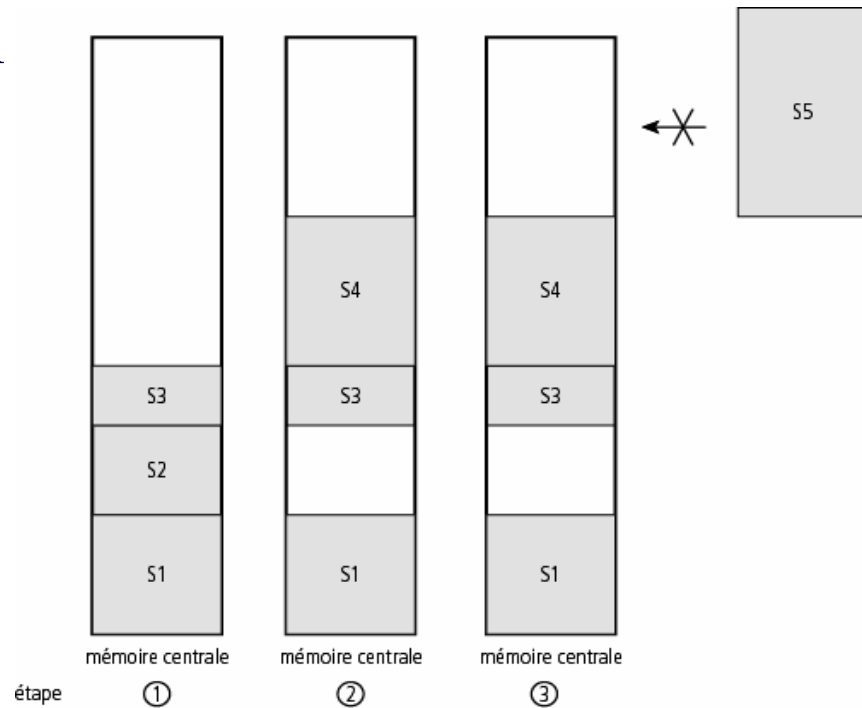


Trois avantages

- Découpage d'un programme en segments
 - présence non obligatoire de tous les segments en MC
 - augmentation du degré de multiprogrammation
- La translation
 - les adresses logiques sont indépendantes de la position du segment en mémoire physique
 - augmentation du degré de multiprogrammation
- La protection
- Le partage par spécialisation du contenu
 - un segment peut être partagé (factorisation du code)
 - code réentrant

Défaut principal

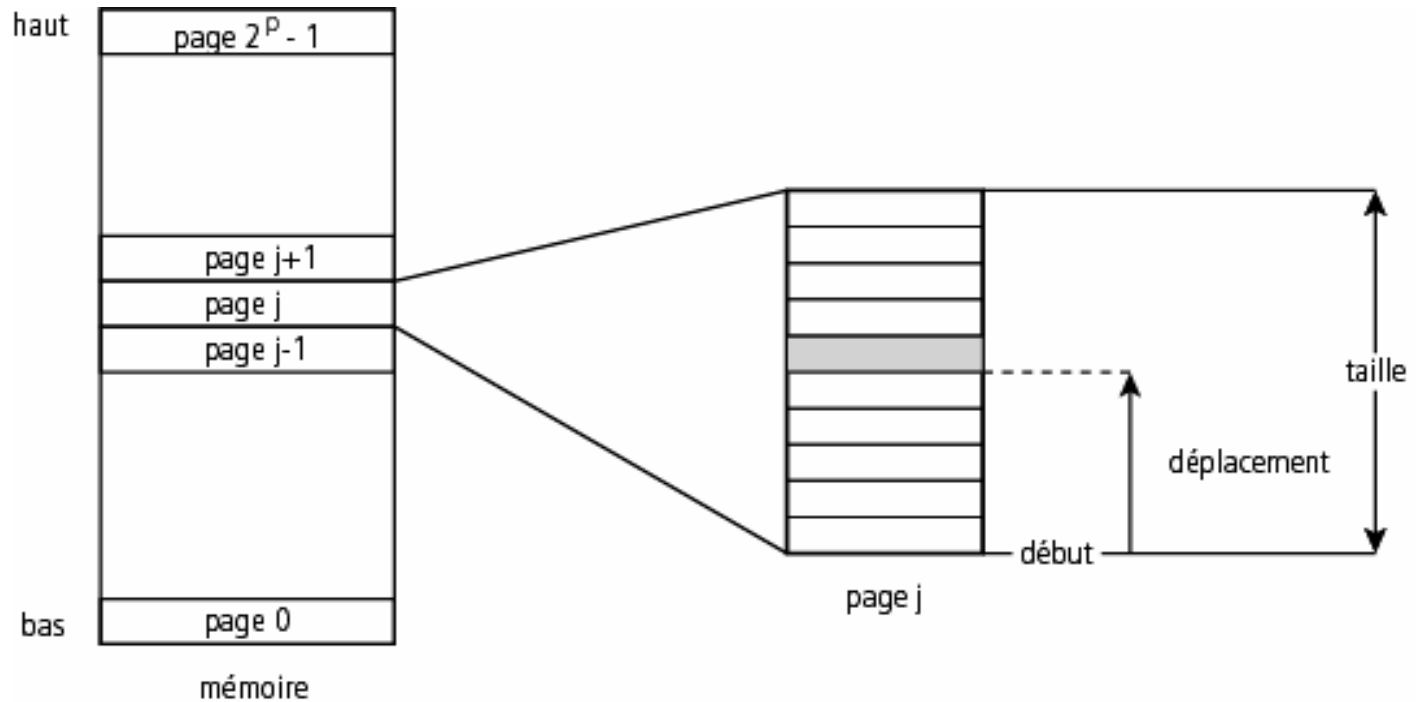
- La fragmentation
- Un scénario



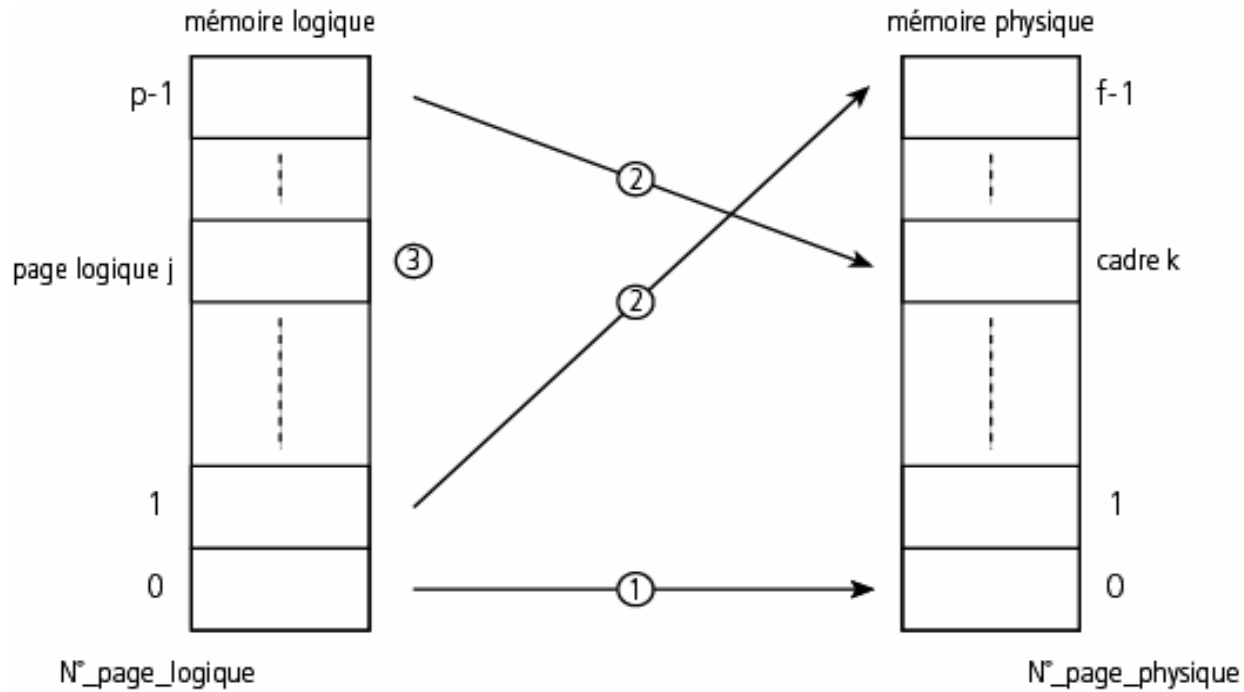
La pagination

- Une page
 - zone contigüe de mémoire
 - taille fixe
 - ordre de grandeur : 1, 4 ou 8 Ko
 - repéré en mémoire par n°_page
 - chaque mot est adressable par rapport au début de la page
 - adressage relatif
 - accueillie dans un cadre en mémoire centrale

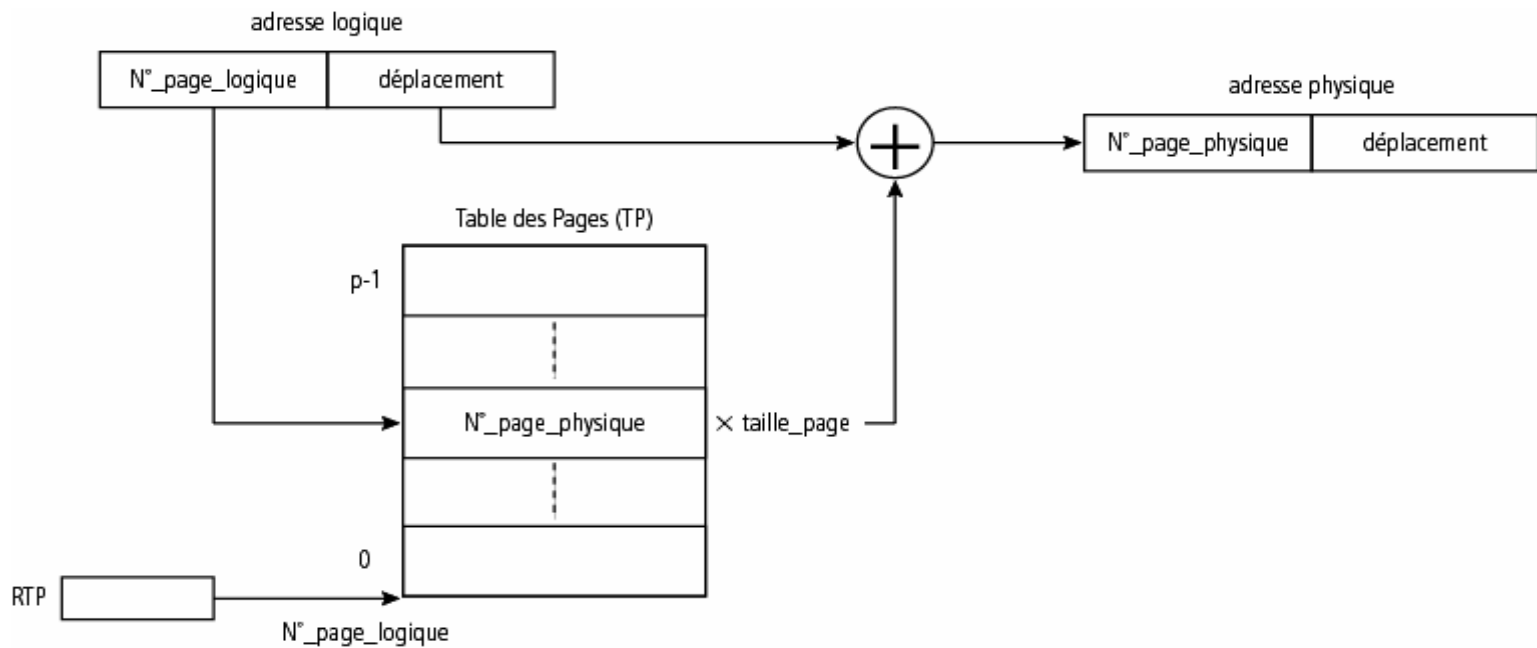
Disposition



Disposition



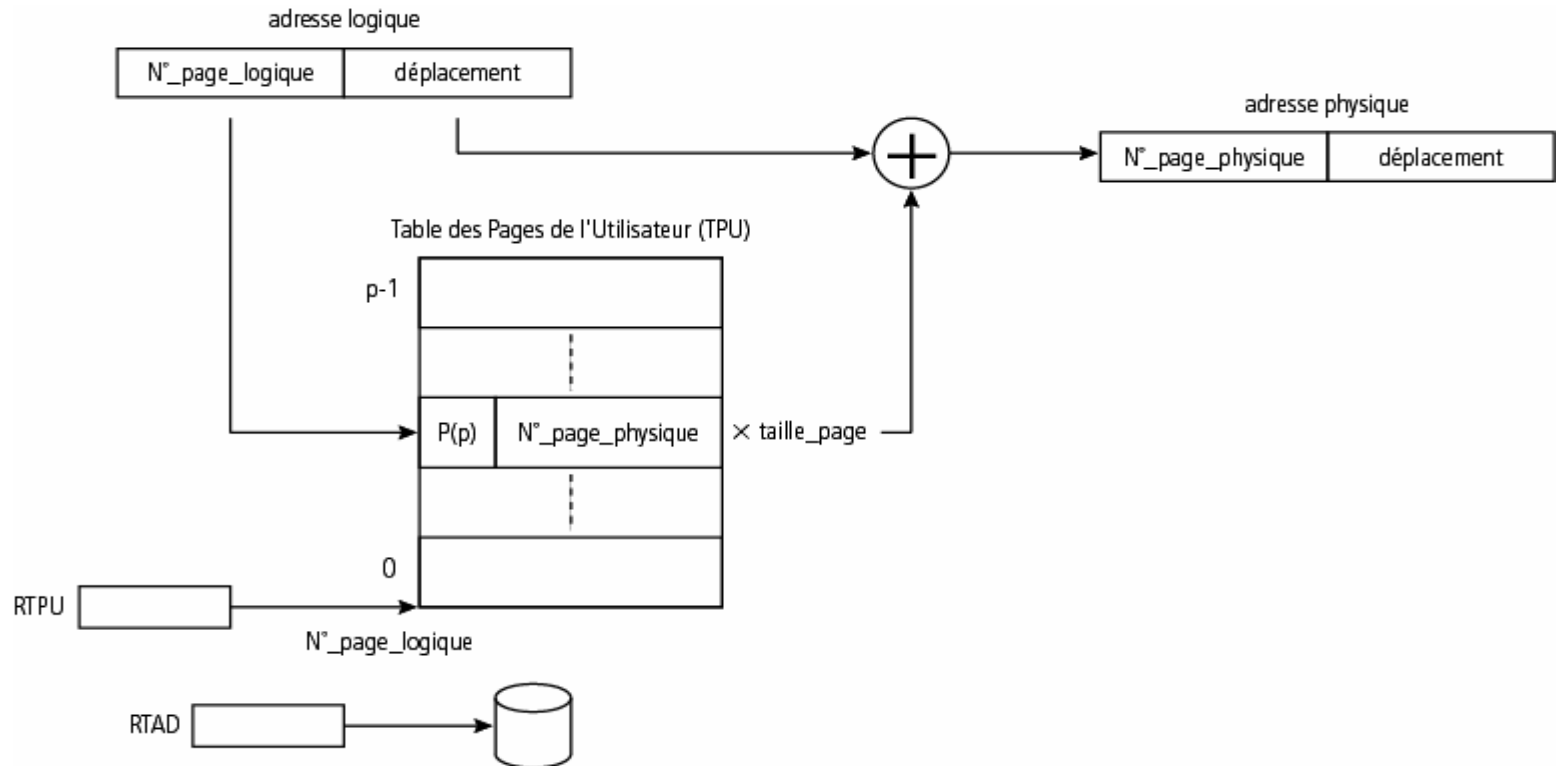
Calcul de l'adresse physique



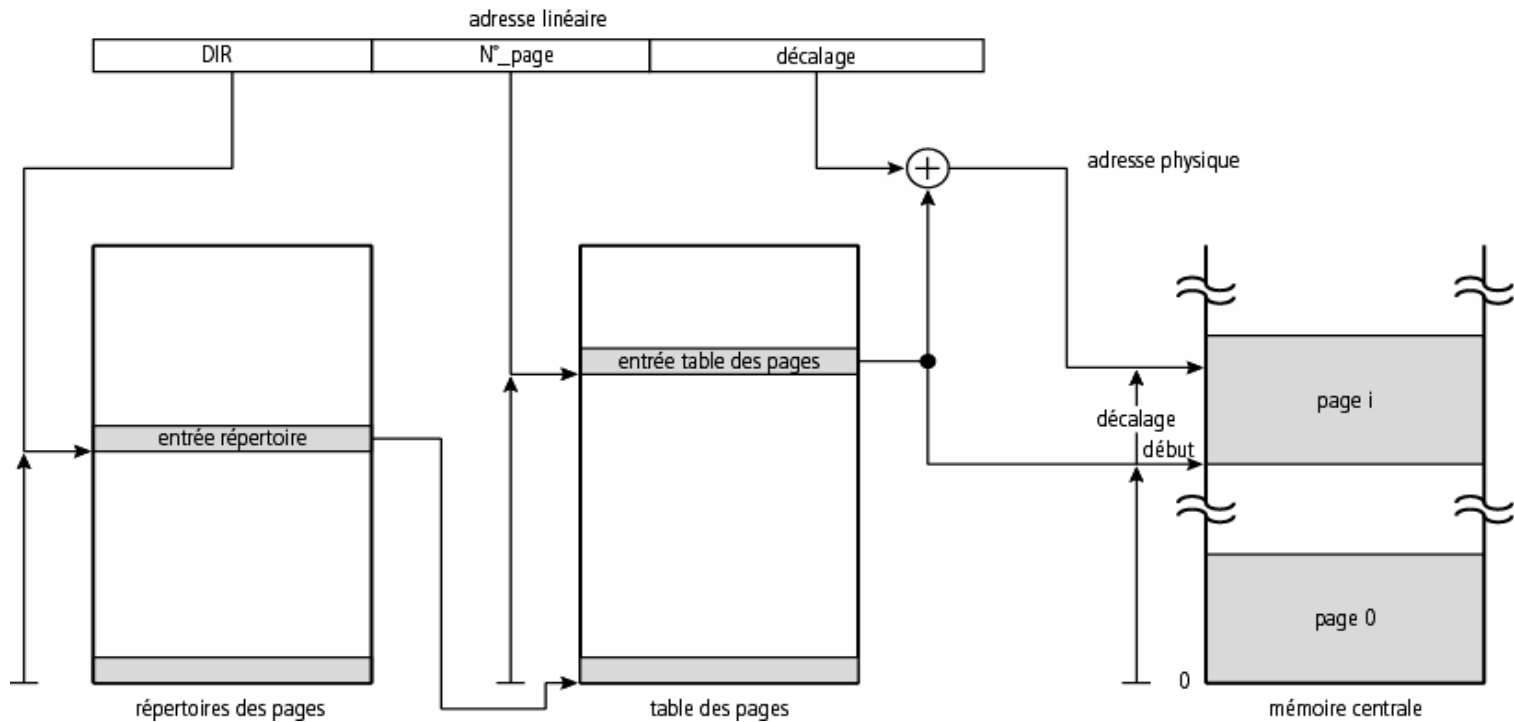
Autres fonctions du MMU

- Si la page physique n'est pas chargée,
 - faute de page (*page fault*)
 - génération d'une interruption par le MMU
- Vérification des droits d'accès
 - lecture/écriture

Mécanisme de protection



Gestion des pages du i386



Trois avantages

- Découpage d'un programme en pages
 - présence non obligatoire de toutes les pages en MC
 - augmentation du degré de multiprogrammation
- Meilleure gestion de la MC
 - problème de la fragmentation de la MC résolu
 - gestion aisée car taille fixe
- La protection

Problèmes à résoudre

□ Problème de la pagination

■ le nombre de pages

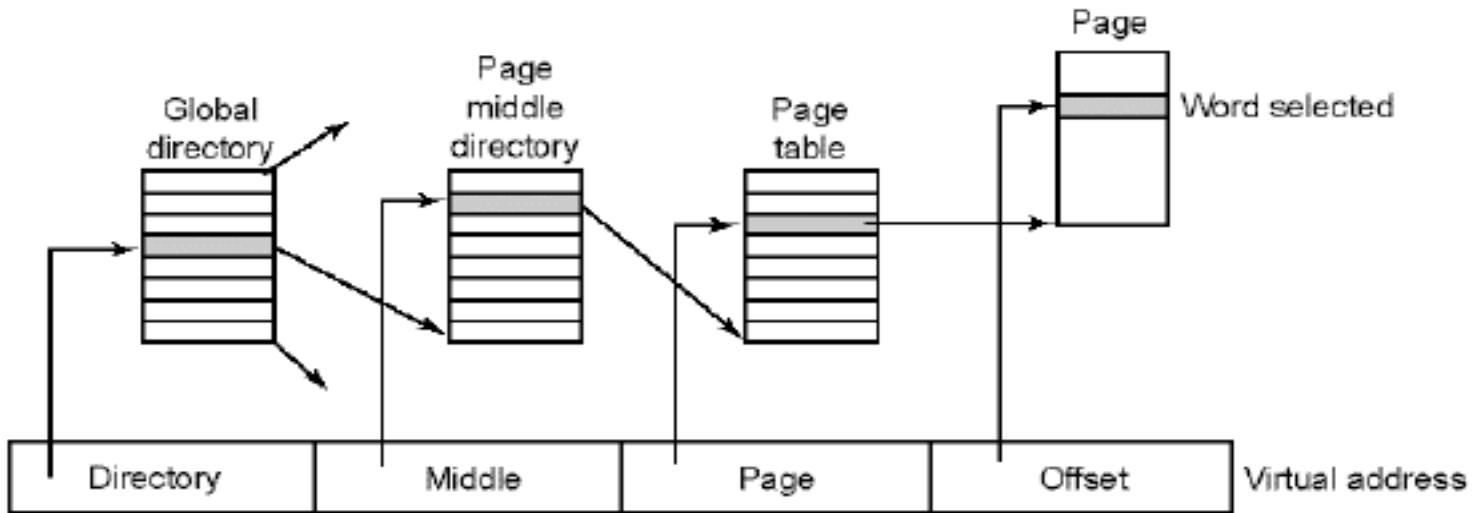
- un espace d'adressage de 32 bits (4 Go) peut représenter 1M pages de 4 Ko

qui implique une très grande table des pages

□ Solutions

- tables des pages hiérarchisées
- utilisation d'un cache
- la pagination inversée

Les tables des pages hiérarchisées



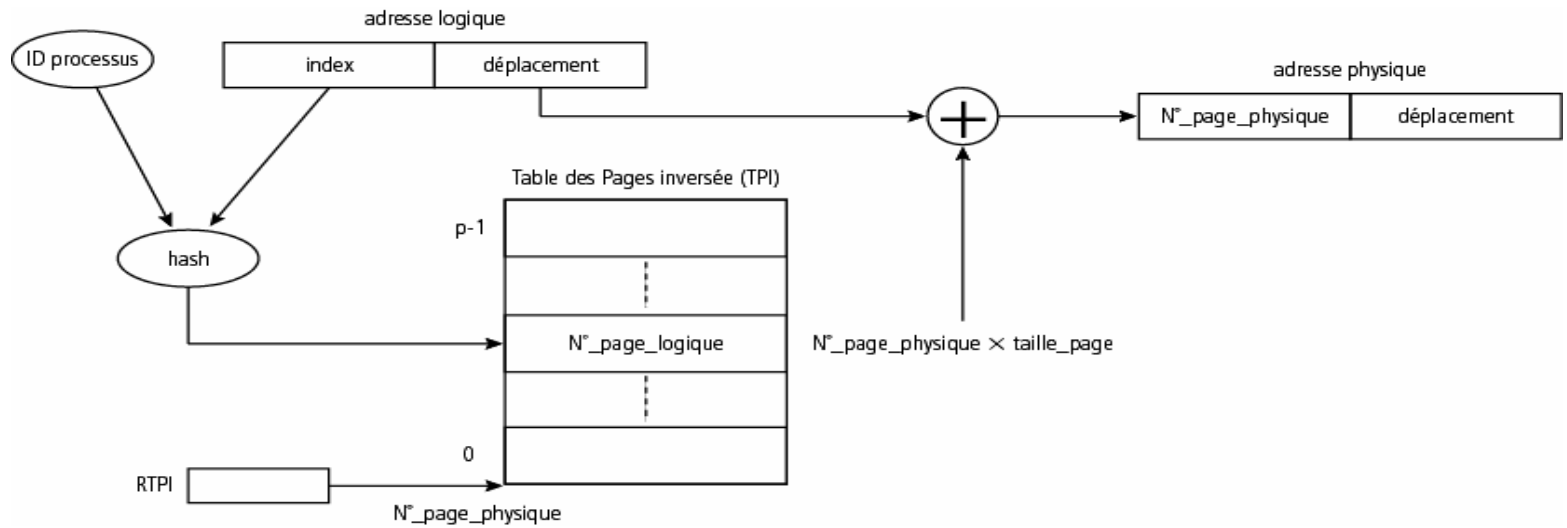
Linux uses three-level page tables



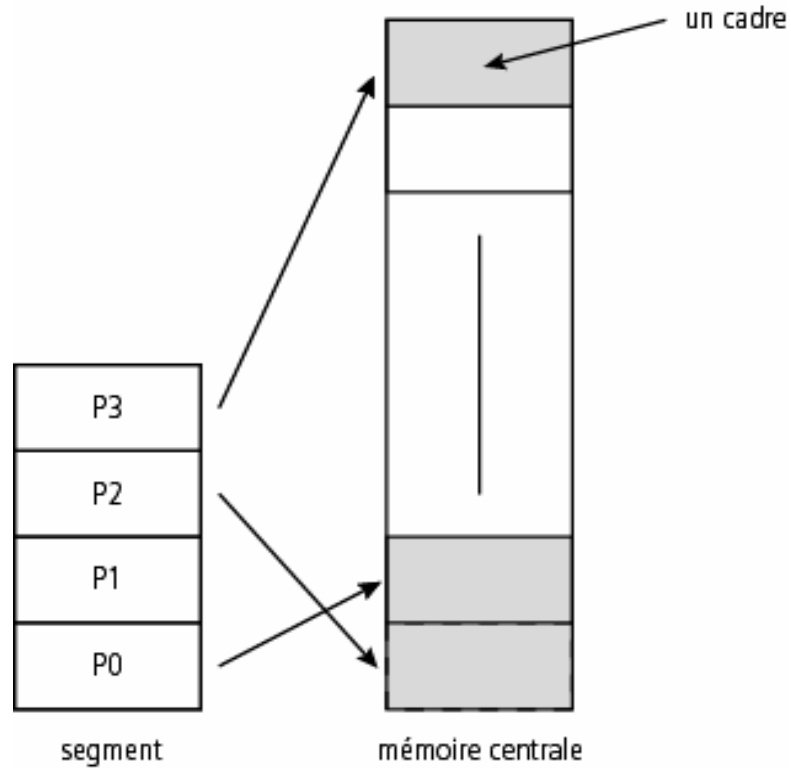
La TLB

- *Translation Look Aside Buffer*
- Utilisation d'un cache

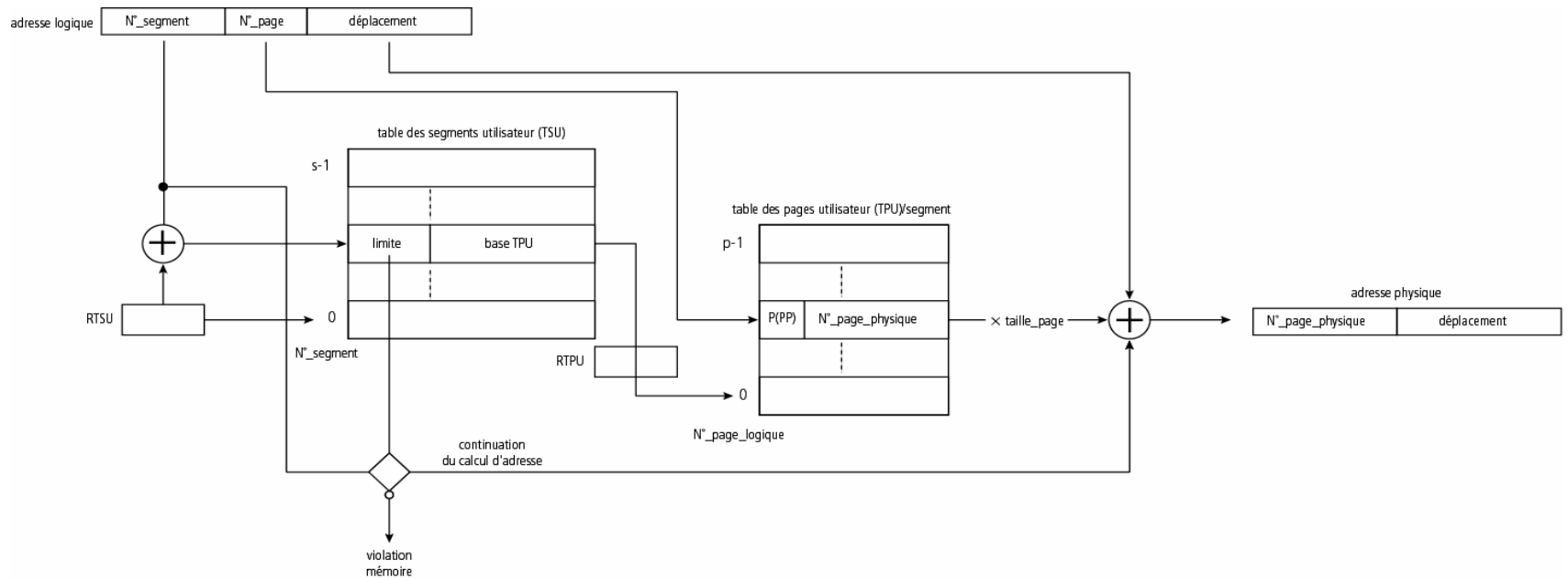
La pagination inversée



Segmentation/pagination



Segmentation/pagination





Rôles du SE

- Chargement
- Remplacement



Conclusion

- XX à faire