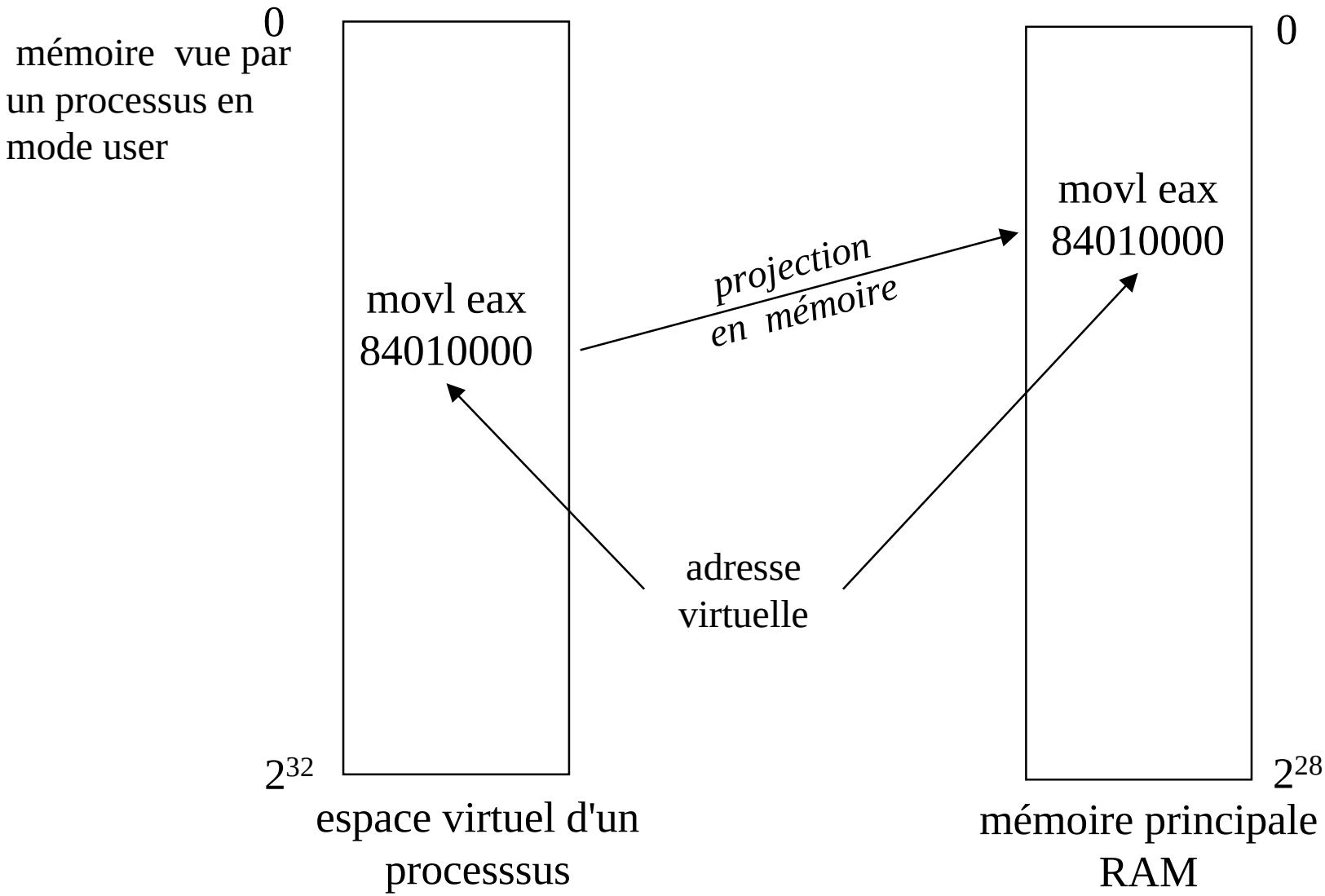
ensiie
école nationale supérieure d'informatique
pour l'industrie et l'entreprise

Systèmes Informatiques
GESTION MEMOIRE

G.BERTHELOT

Mémoire d'un processus



Mémoire d'un processus

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- inconvénient : cette traduction prend du temps et ralentit à chaque fois le processeur
- elle n'est supportable que si le processeur offre les moyens (MMU) pour faire ce travail sans trop pénaliser
- avantage : indépendance de la mémoire physique et de l'espace d'adresses virtuelles
- isolation des processus entre eux

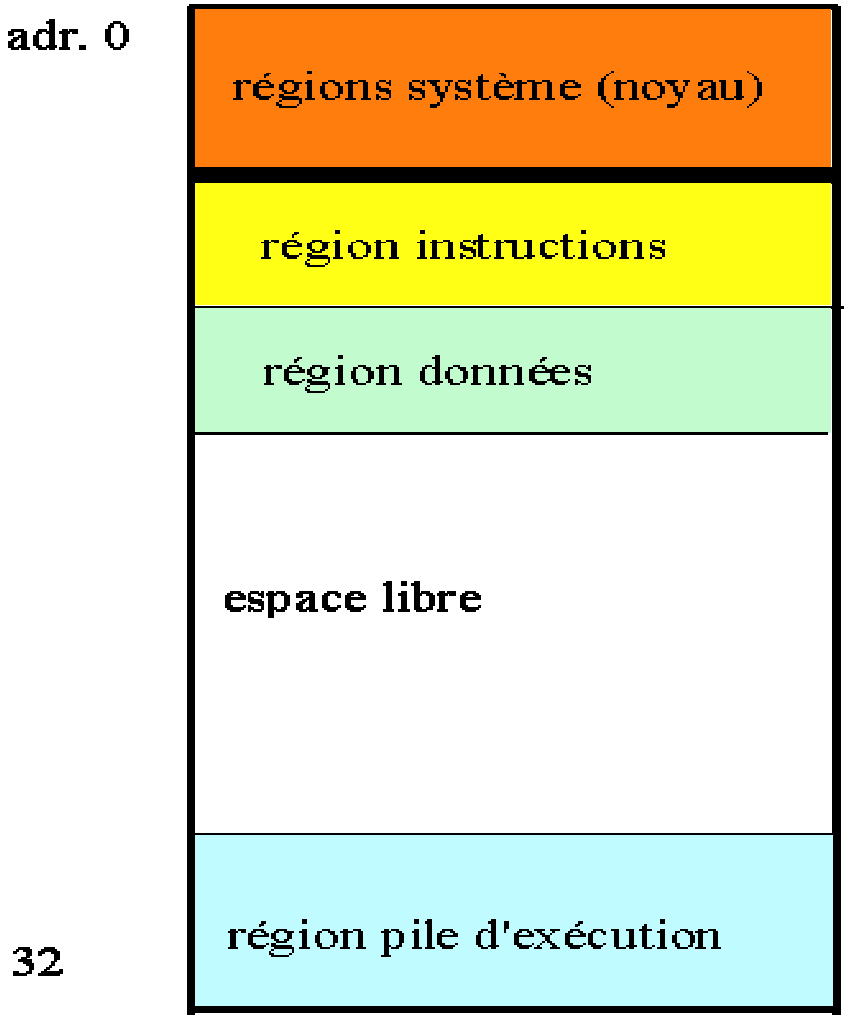
Mémoire d'un processus

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- inconvénient : difficile de partager
  - instructrions
  - données

Mémoire d'un processus

La mémoire virtuelle est structurée en régions



Mémoire d'un processus

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- la taille de chacune des régions est fixée en fonction du programme exécutable
- si un programme essaye d'accéder à une adresse en dehors d'une région -> *segmentation fault*

  (normal traduction d'adresse impossible)
- certaine régions peuvent être augmentées (*tas pile*) mais après demande au système

Mémoire d'un processus

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Les régions peuvent être placées à d'autres endroits si la demande en est faite lors de la construction de l'exécutable

- les régions inutilisées ne sont pas gérées

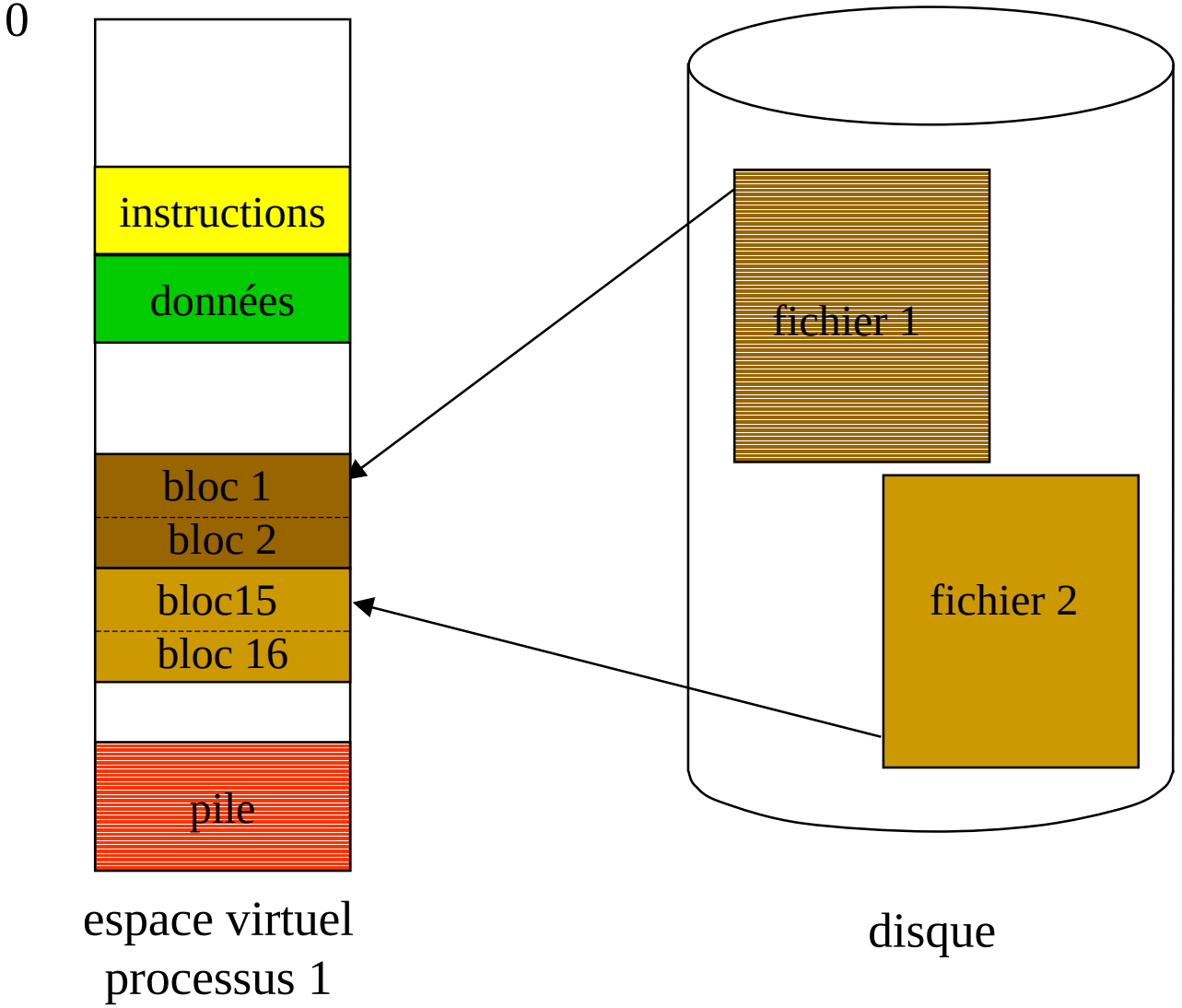
  (image mémoire d'un processus)

Mémoire d'un processus

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- d'autres régions peuvent être ajoutées
  - ajout d'instructions et de données en cours d'exécution (dlopen) (augmentation du programme)
  - projection de fichier en mémoire (accès direct au contenu d'un fichier sans passer par read ou fread)
  - segments de mémoire partagés

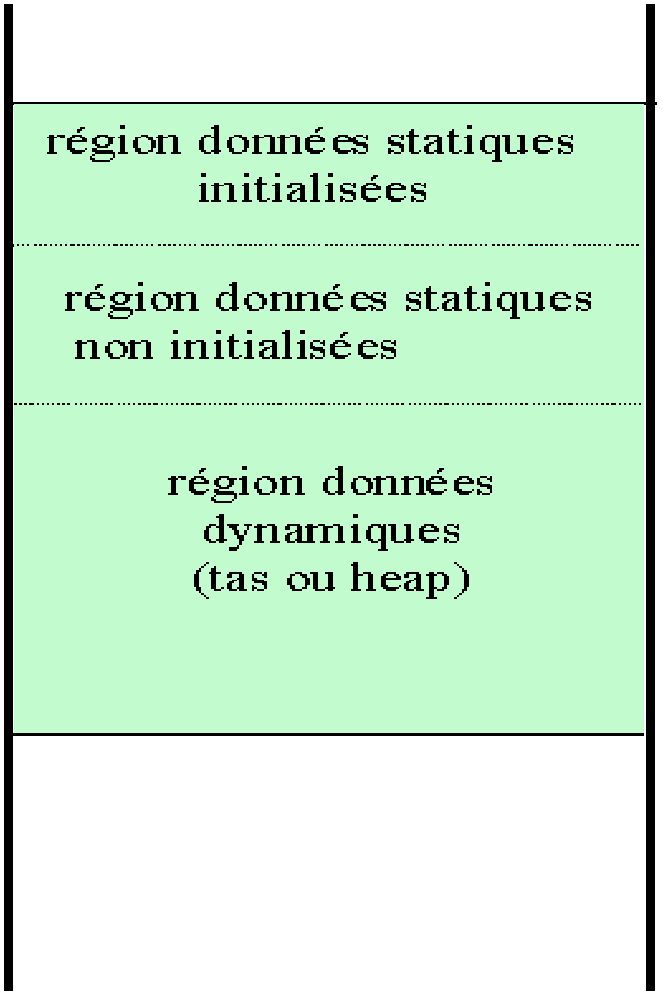
Projection de fichiers (mapping)



Mémoire virtuelle d'un processus

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

Régions données



Remarque :
le tas est géré à l'aide de la fonction malloc()

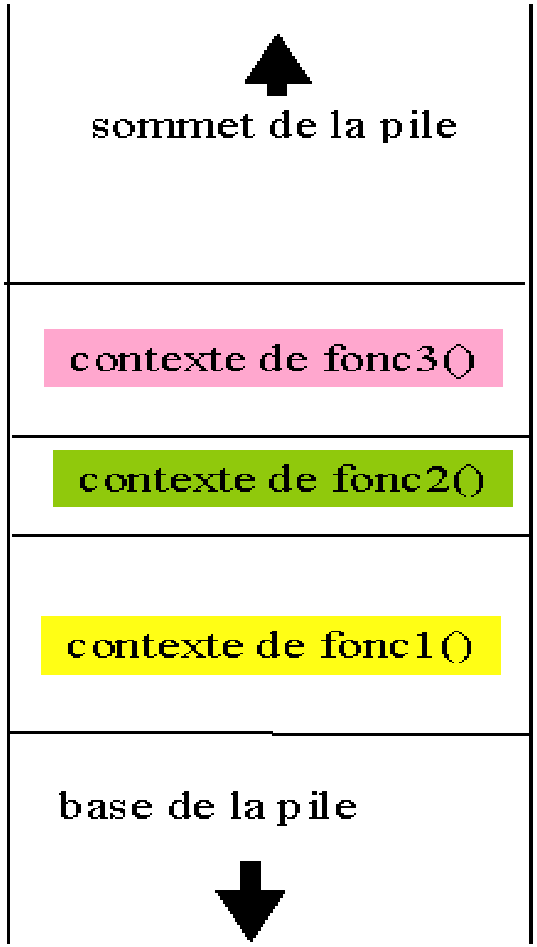
Mémoire d'un processus

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

Régions pile

en cours d'exécution : fonc3()

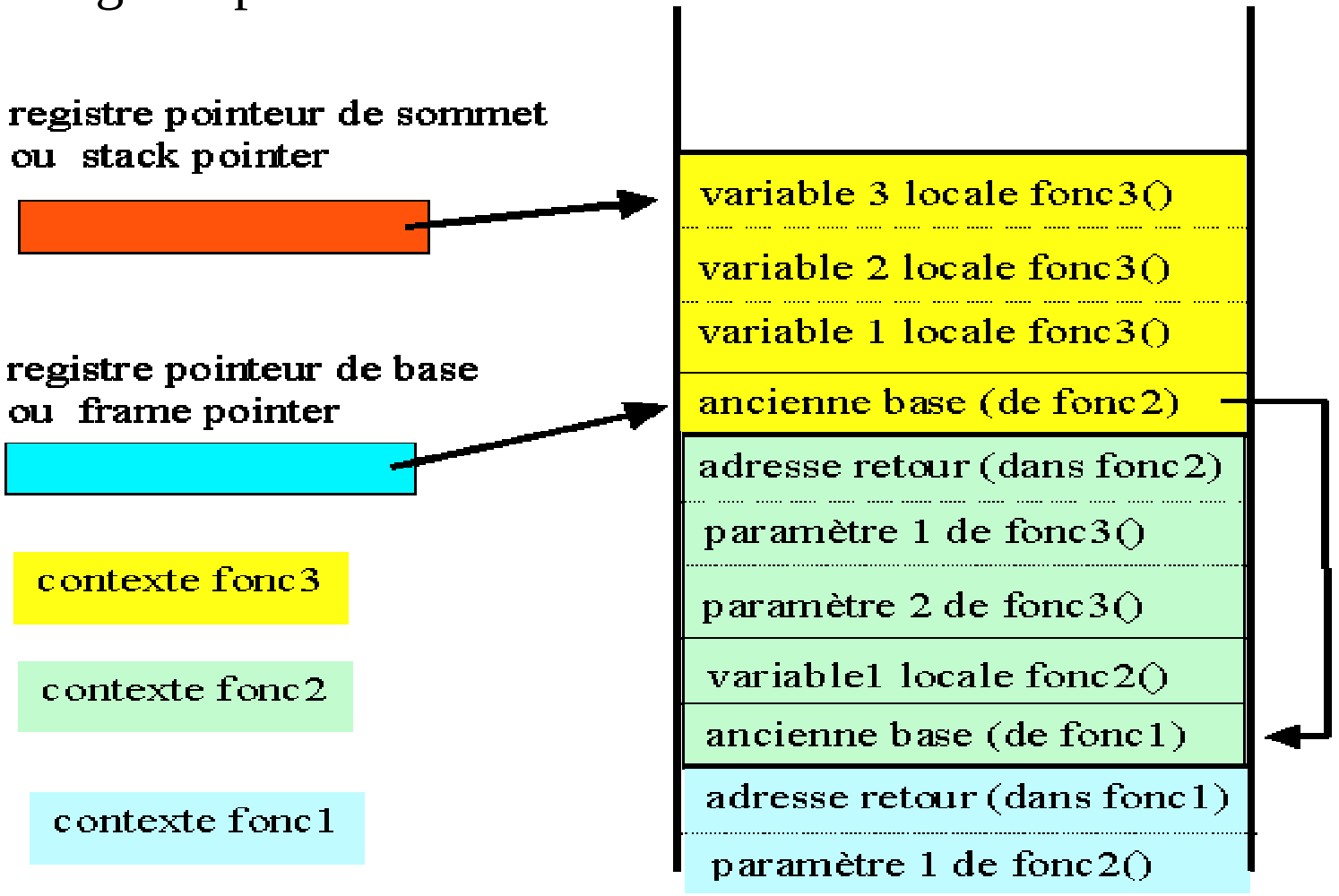
```
void fonc1(...){
..
fonc2(...);
..
};

void fonc2(...){
..
fonc3(...);
..
}

void fonc3(...){
..
..
..
};
```



Mémoire d'un processus

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

Régions pile

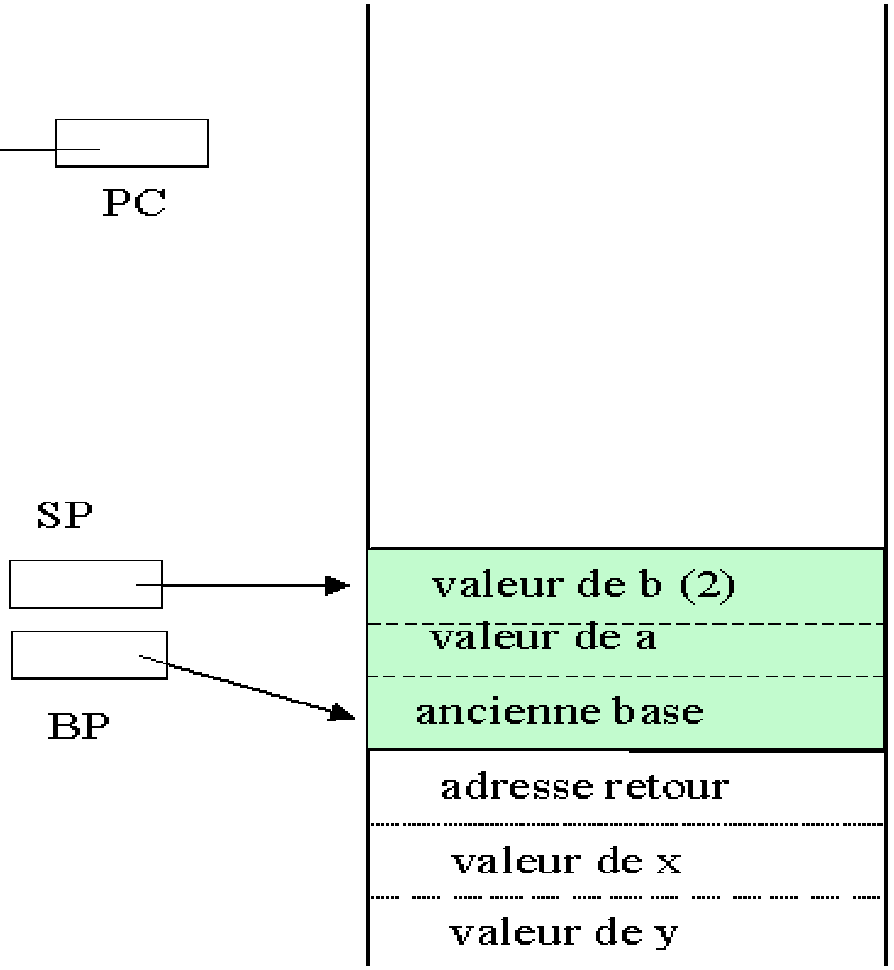


Mémoire d'un processus

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

Régions pile

```
void princ (int x,y){
int a,b=2;
a = som(x,y);
return a/b;
};

int som(g,h){
int z;
z = g+h;
return z;
}
```

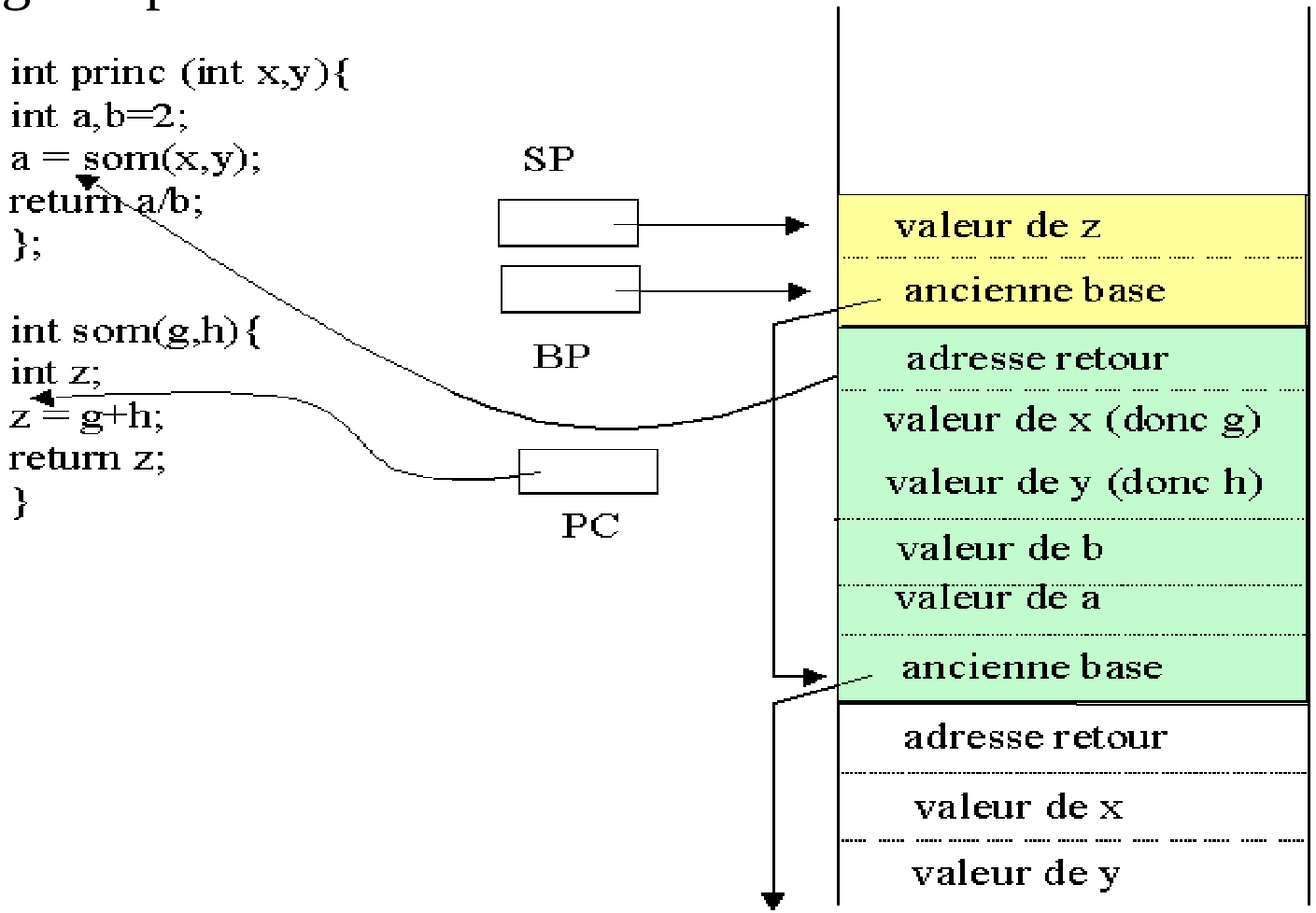


Mémoire d'un processus

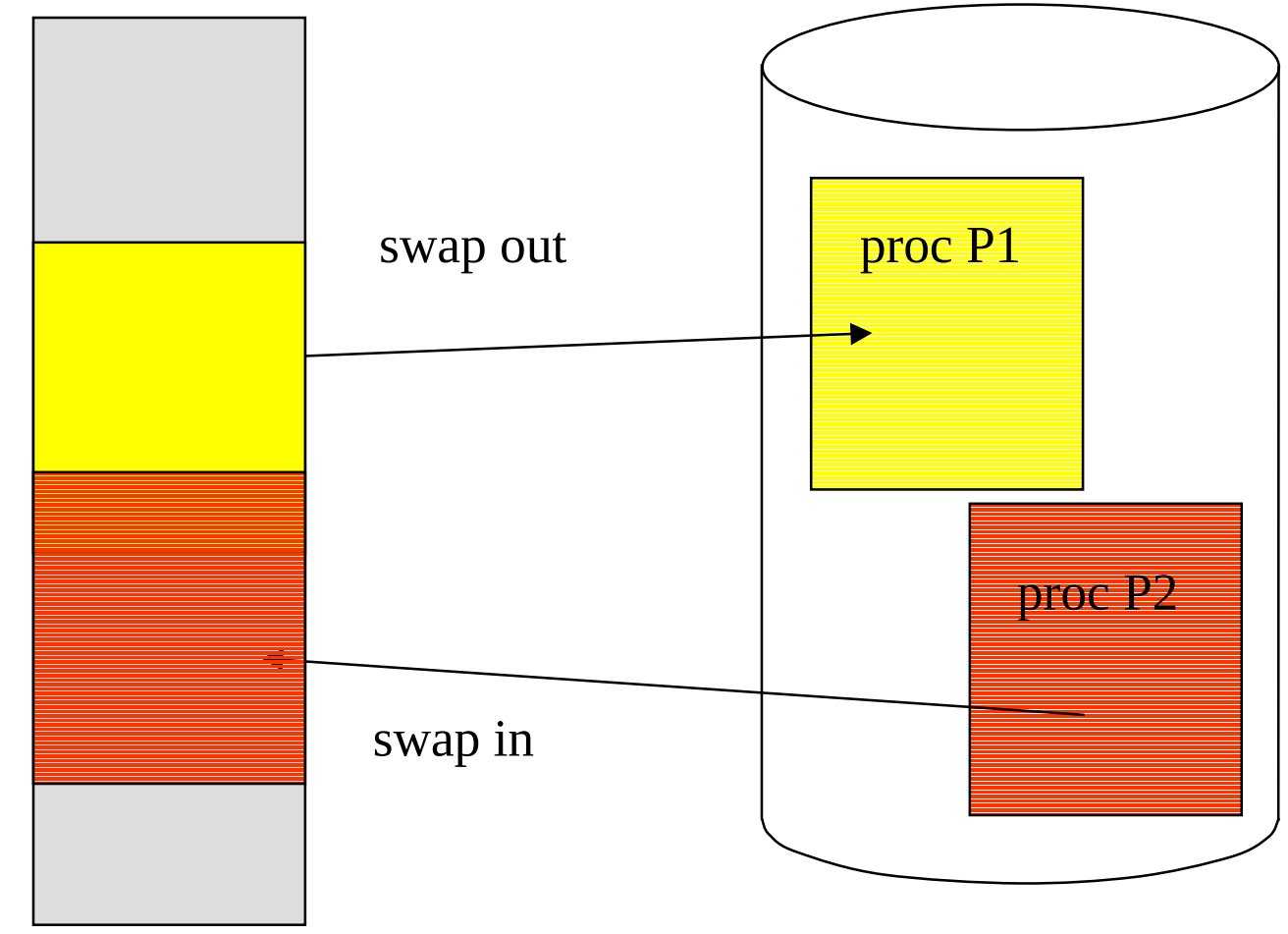
ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

Régions pile

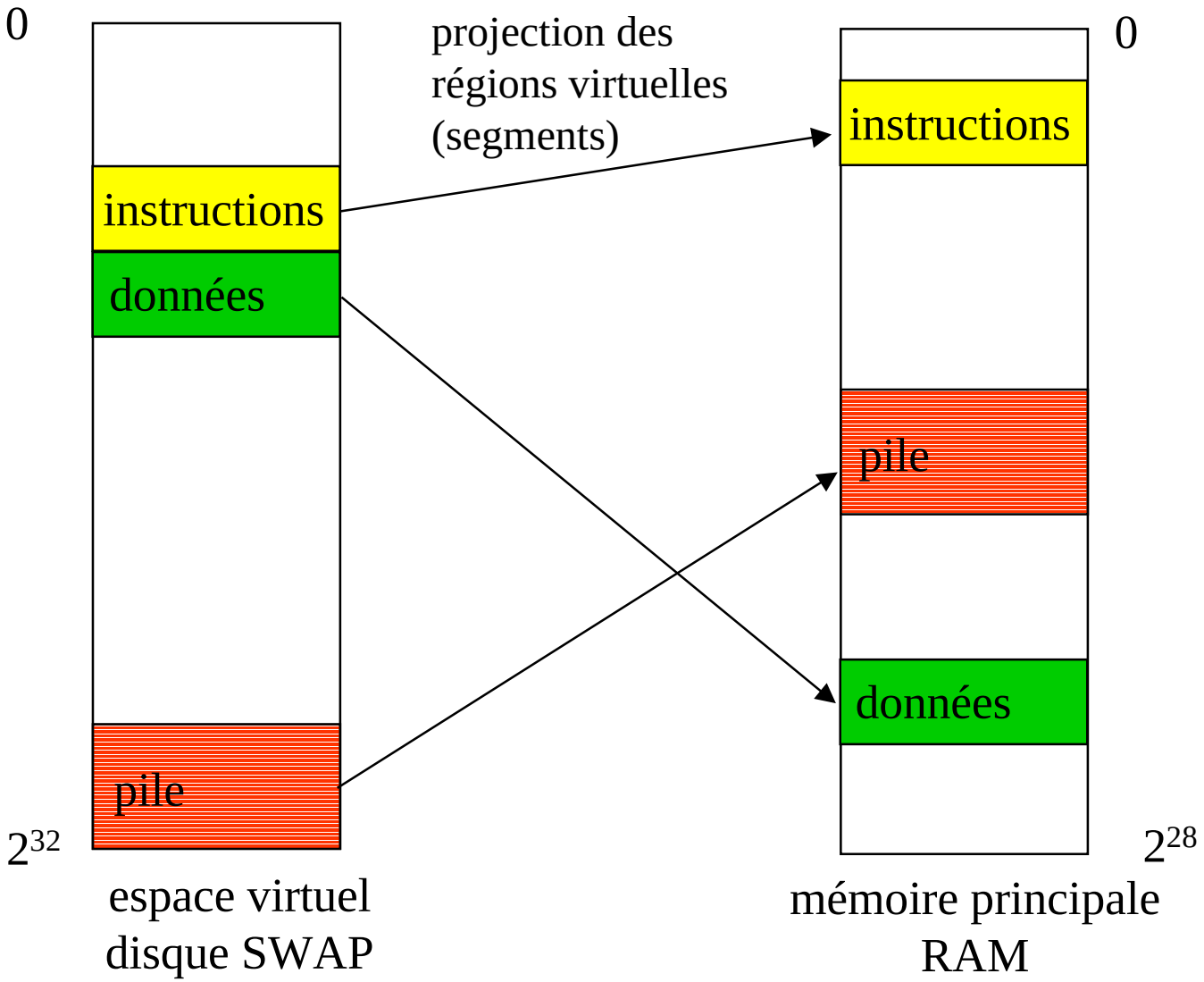
```
int princ (int x,y){
int a,b=2;
a = som(x,y);
return a/b;
};

int som(g,h){
int z;
z = g+h;
return z;
}
```



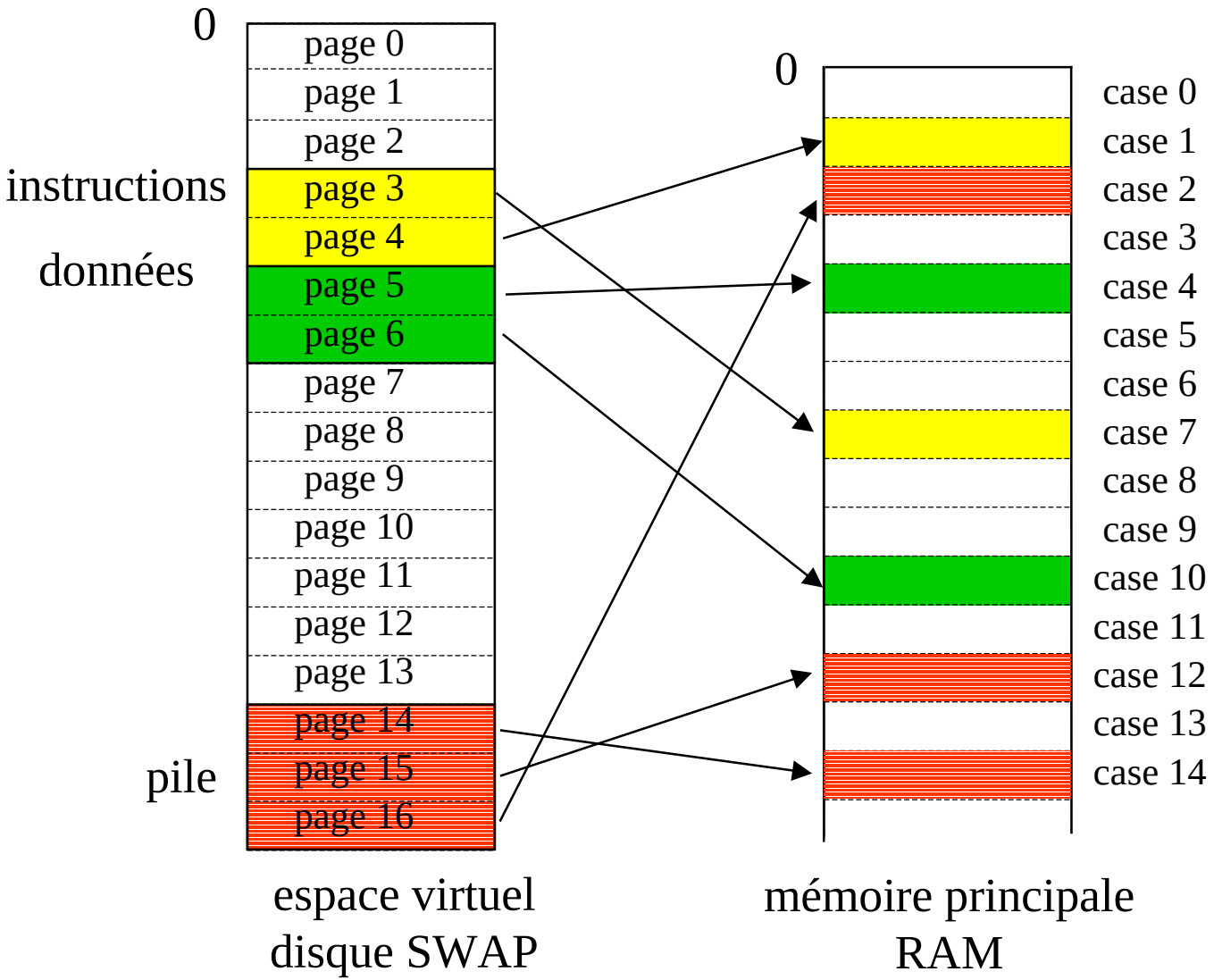
Mémoire d'un processus



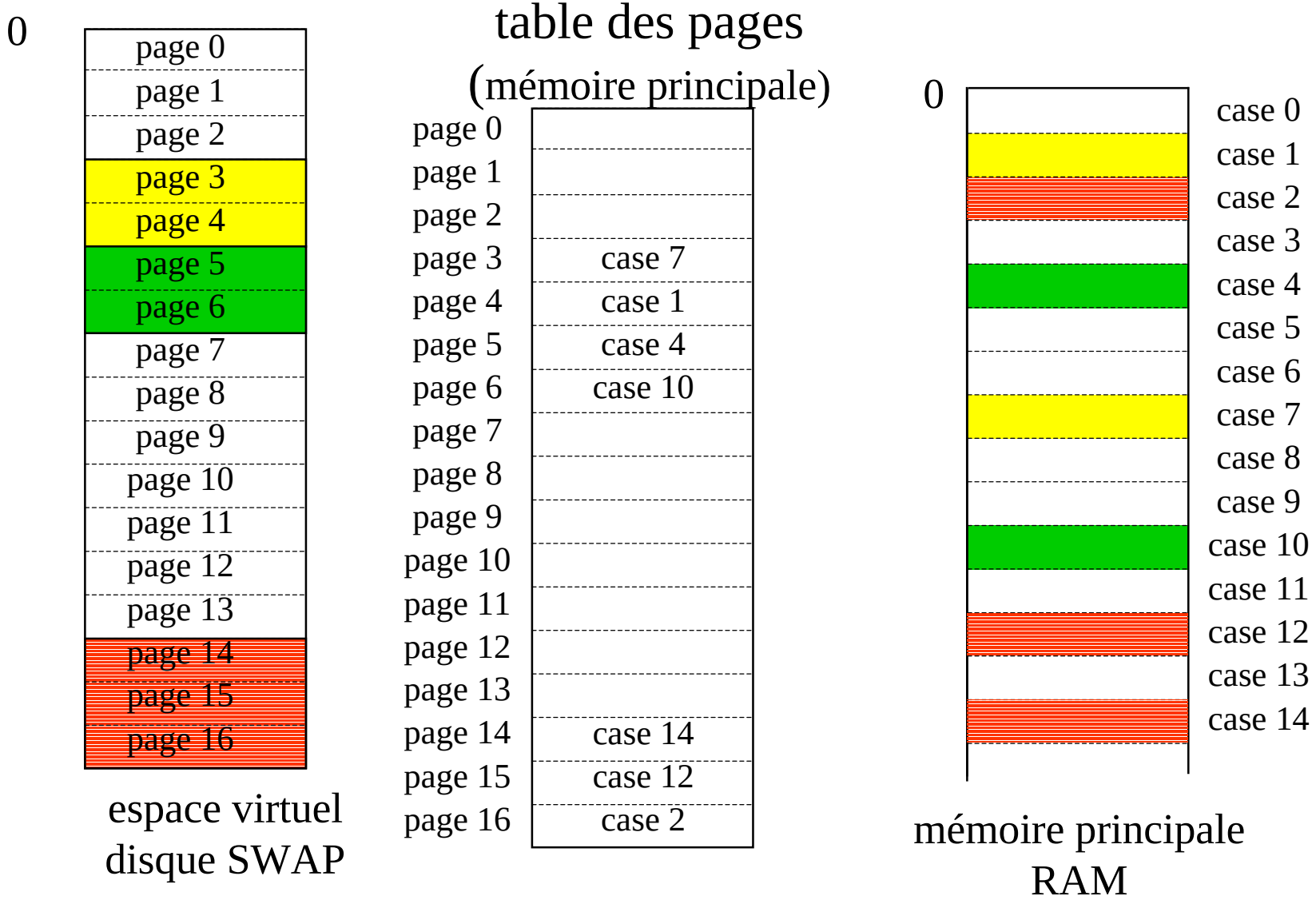
Mémoire d'un processus



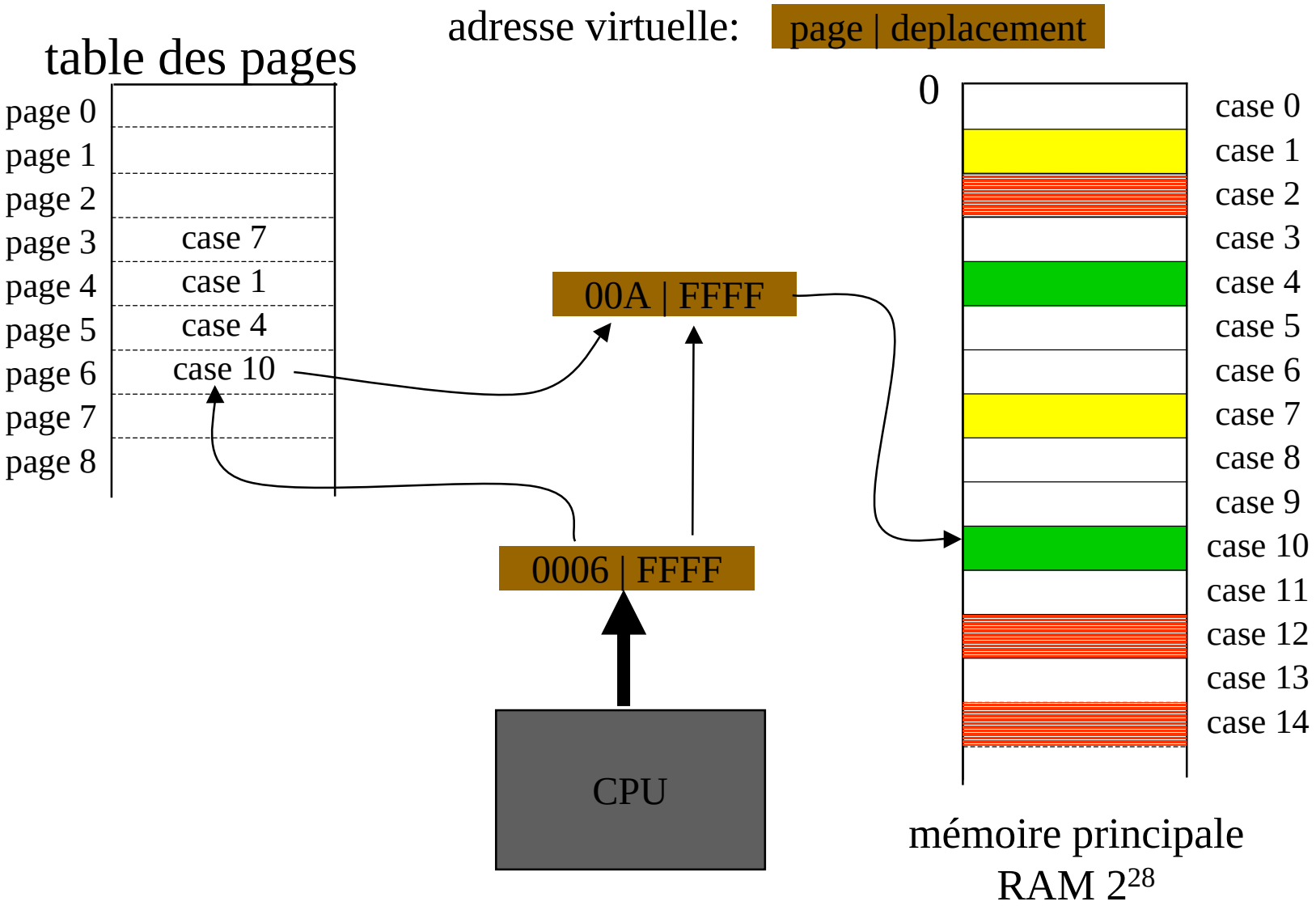
pagination:
taille page = taille case



pagination


0
page 0
page 1
page 2
page 3
page 4
page 5
page 6
page 7
page 8
page 9
page 10
page 11
page 12
page 13
page 14
page 15
page 16
espace virtuel
disque SWAP
table des pages
(mémoire principale)
page 0
page 1
page 2
page 3
page 4
page 5
page 6
page 7
page 8
page 9
page 10
page 11
page 12
page 13
page 14
page 15
page 16
case 7
case 1
case 4
case 10
case 14
case 12
case 2
0
case 0
case 1
case 2
case 3
case 4
case 5
case 6
case 7
case 8
case 9
case 10
case 11
case 12
case 13
case 14
mémoire principale
RAM

pagination :
Traduction des adresses



Pagination

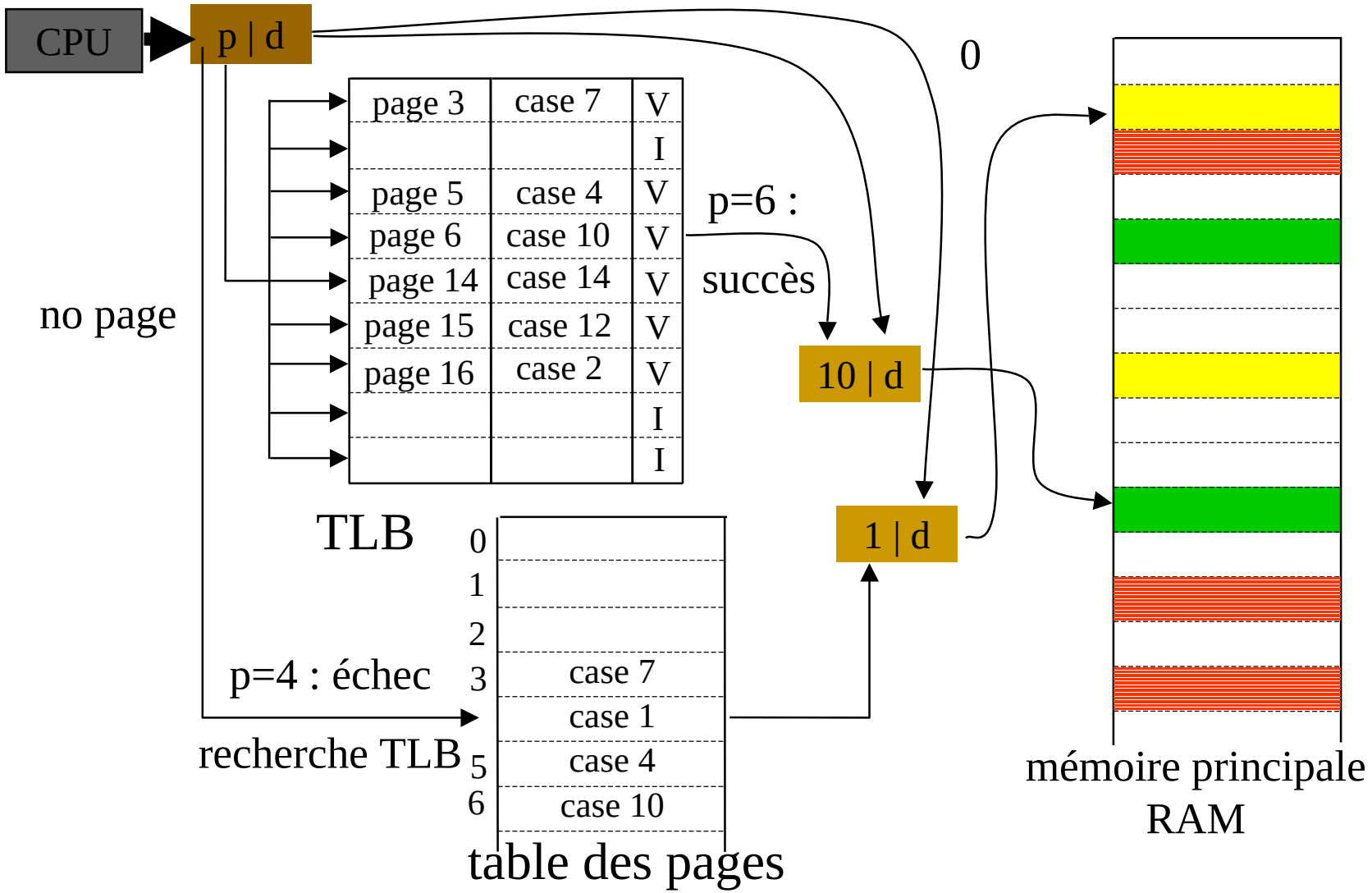
ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- problème :
  - taille de la table

    Exemple avec 32bits d'adresse virtuelle et des pages de 4K (12 bits): 2^{20} entrées de 4 octets soit 8MO par table
  - durée de la recherche
  - registre de la table des pages changé à chaque commutation

- TLB (cache de la table des pages)

pagination :
Translation Look-aside Buffer



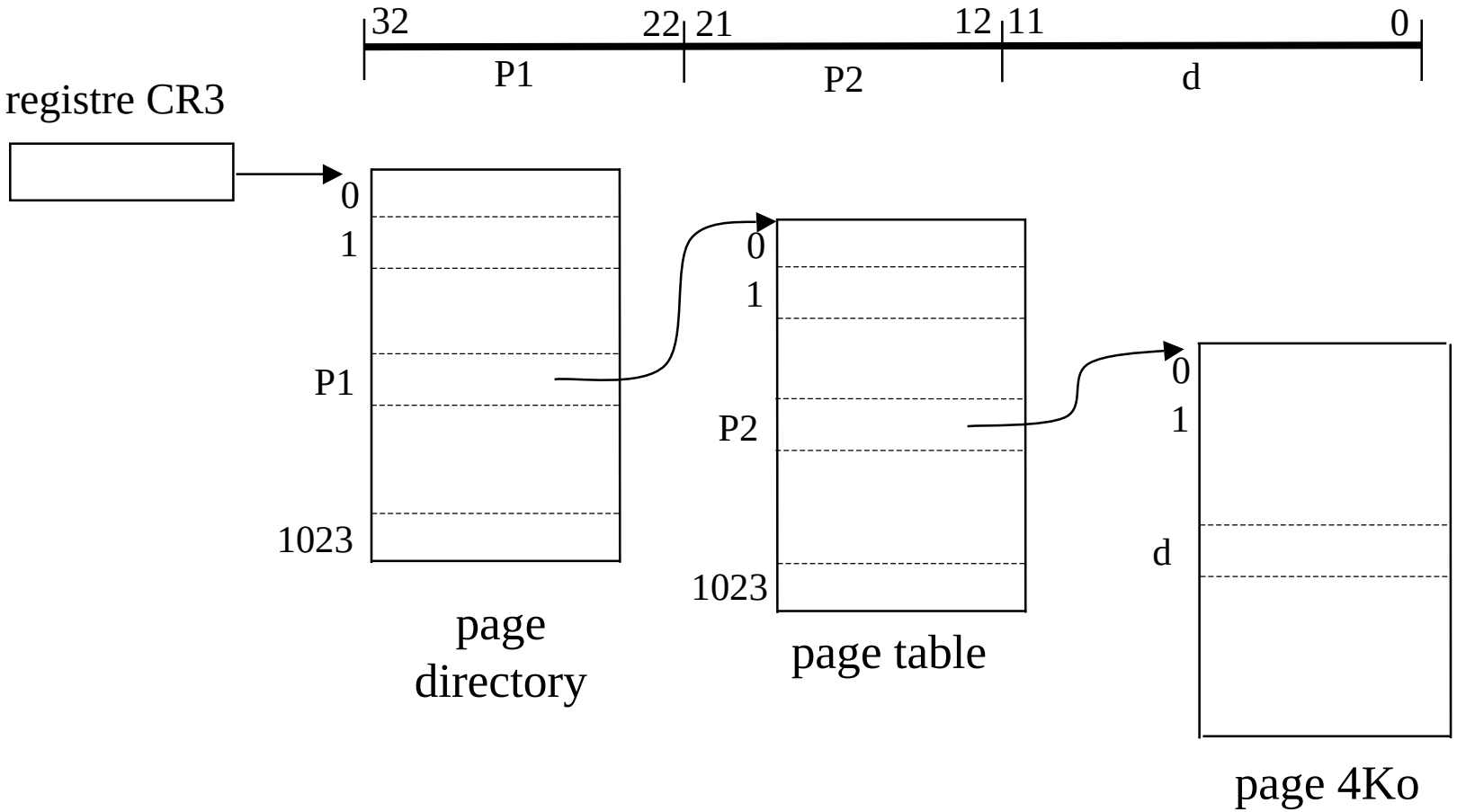
Pagination

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

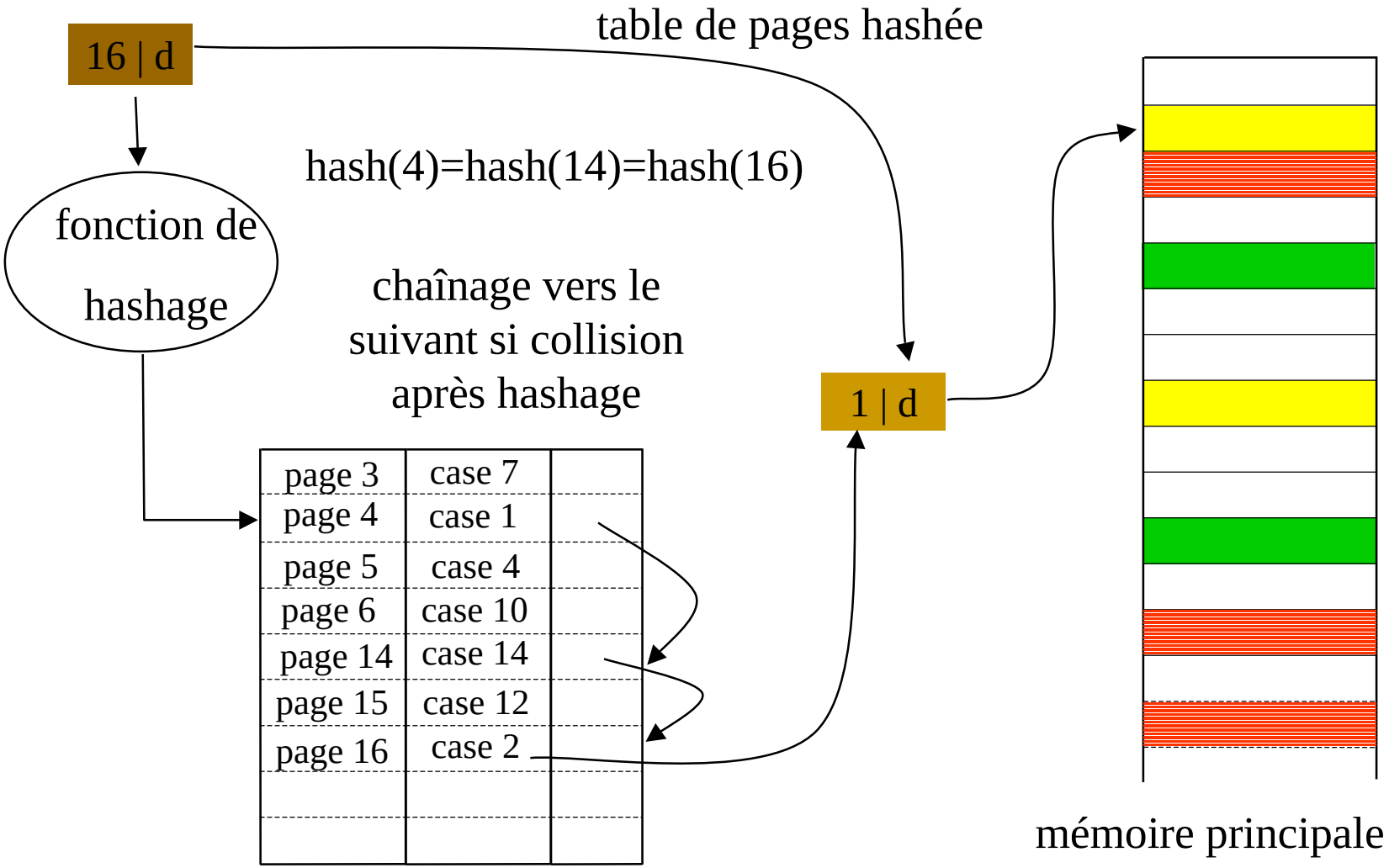
- TLB entre 64 et 1024 entrées
  - taux de succès : calcul du ralentissement
  - gestion des renouvellements
  - les bits de validité permettent de ne pas renouveller le TLB en totalité à chaque commutation
- dispositifs matériels : Mémory Management Unit

Pagination

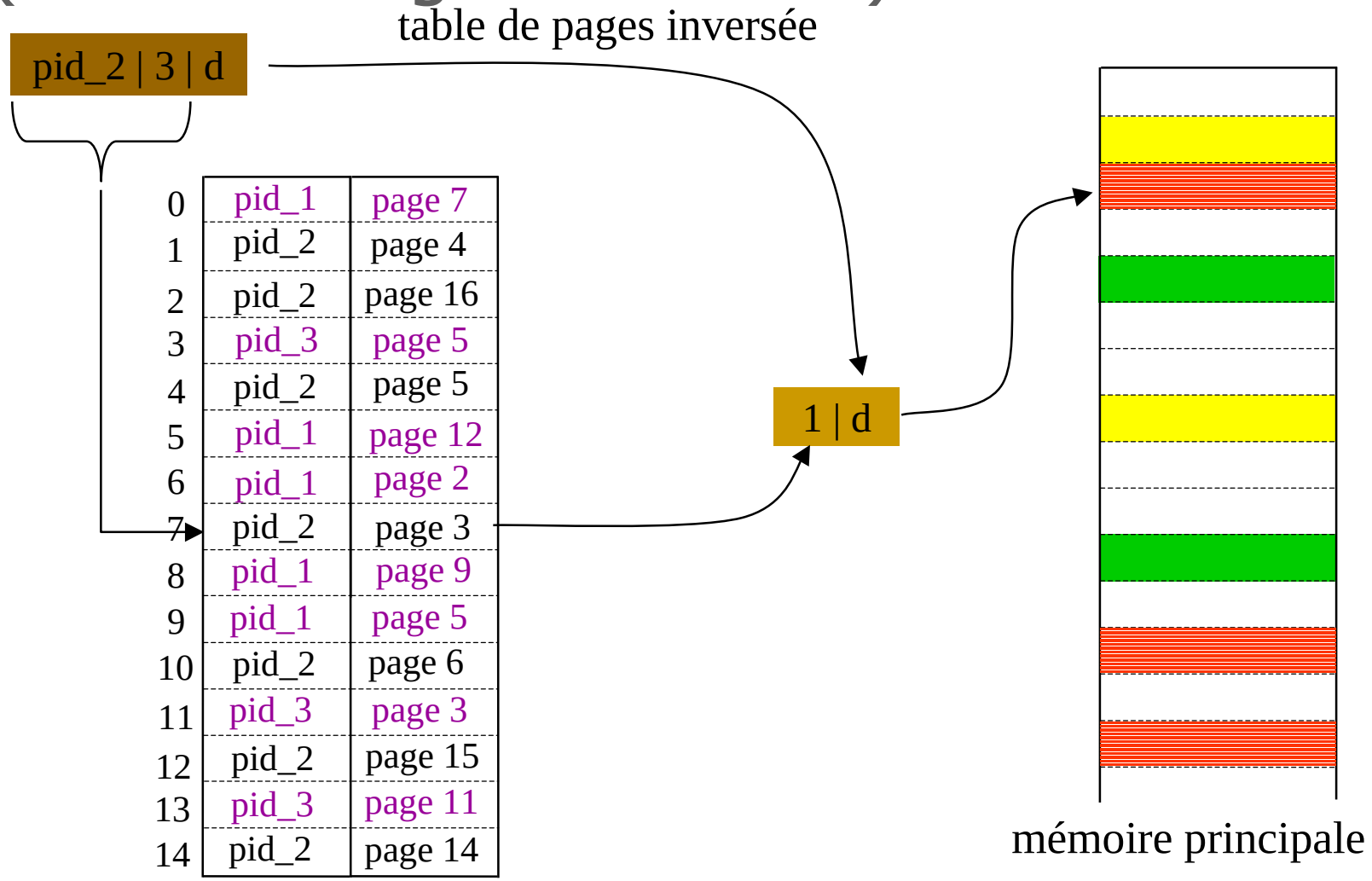
- Table des pages hiérarchisées



Pagination
(mémoires grande taille)



Pagination
(mémoires grande taille)



une entrée par case, recherche linéaire après TLB

Partage de régions

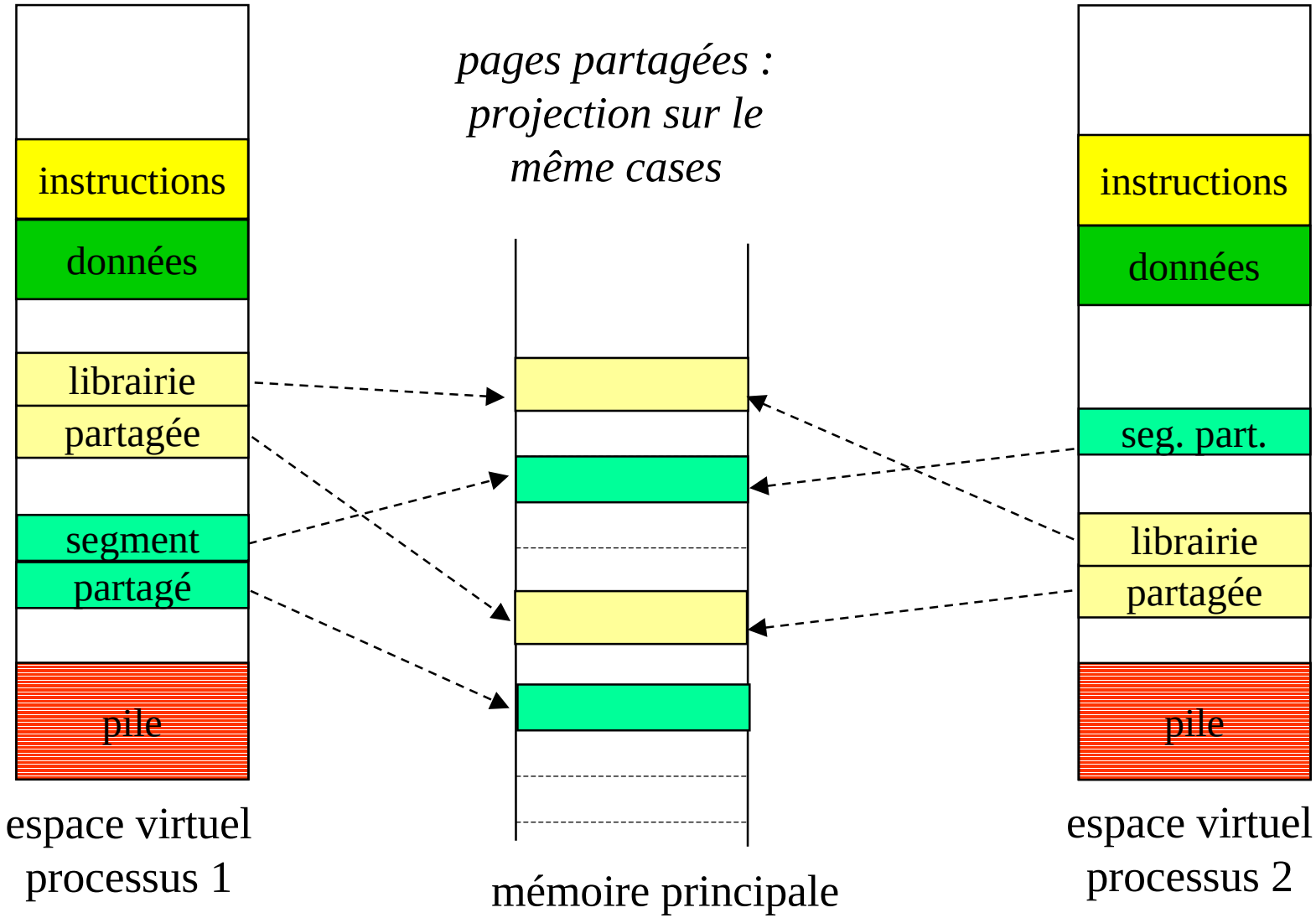
ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- partages d'instruction (code)
  - librairies (en particulier libraires système)
    - avantage :

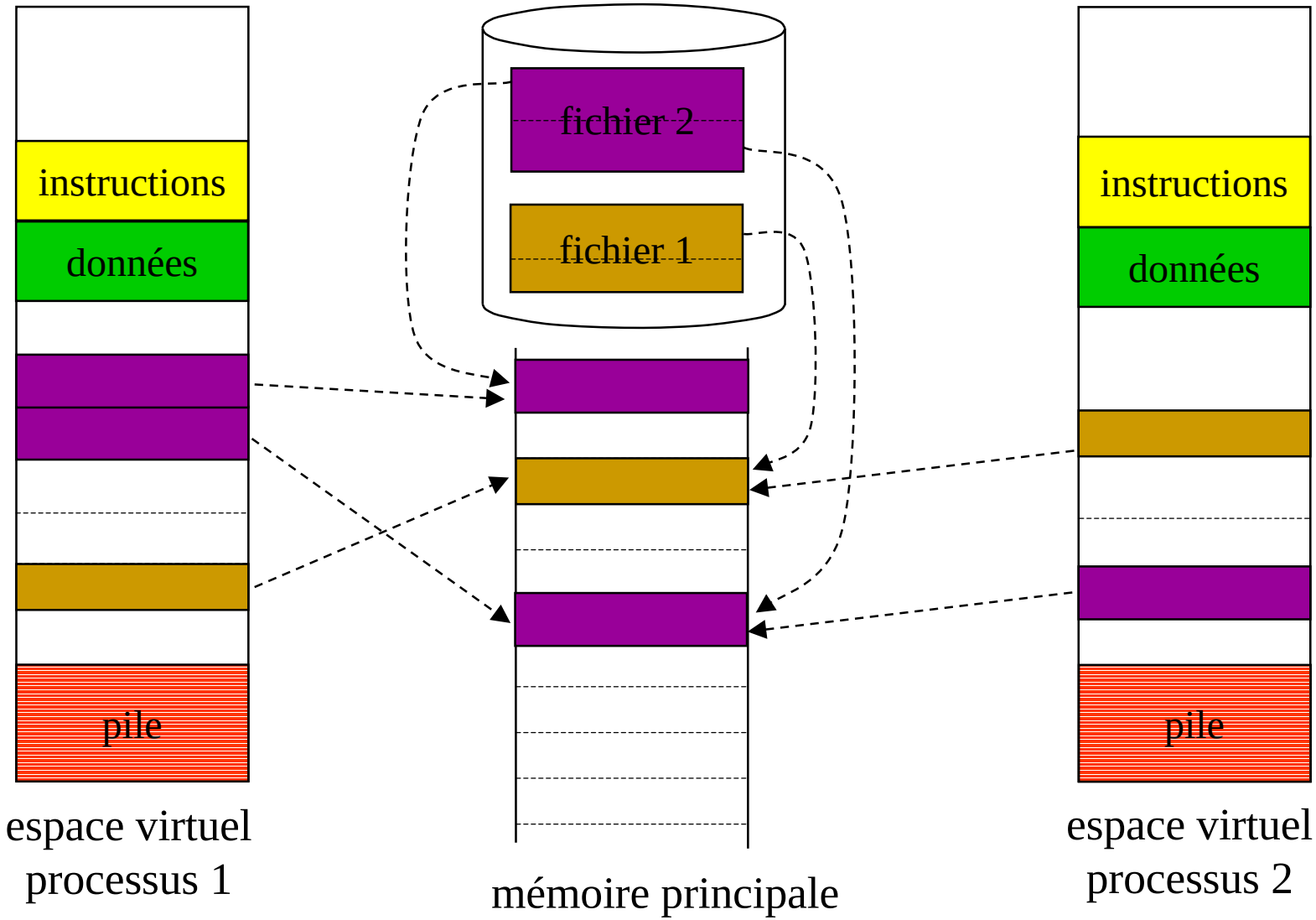
évite d'encombrer la mémoire principale et la mémoire de réserve avec des pages ayant les mêmes contenus.

    - Restrictions
      - code réentrant : en contient pas de variables et ne se modifie pas en cours d'éxécutiun
      - code position indépendant

bibliothèques et segments partagés



Fichiers projetés



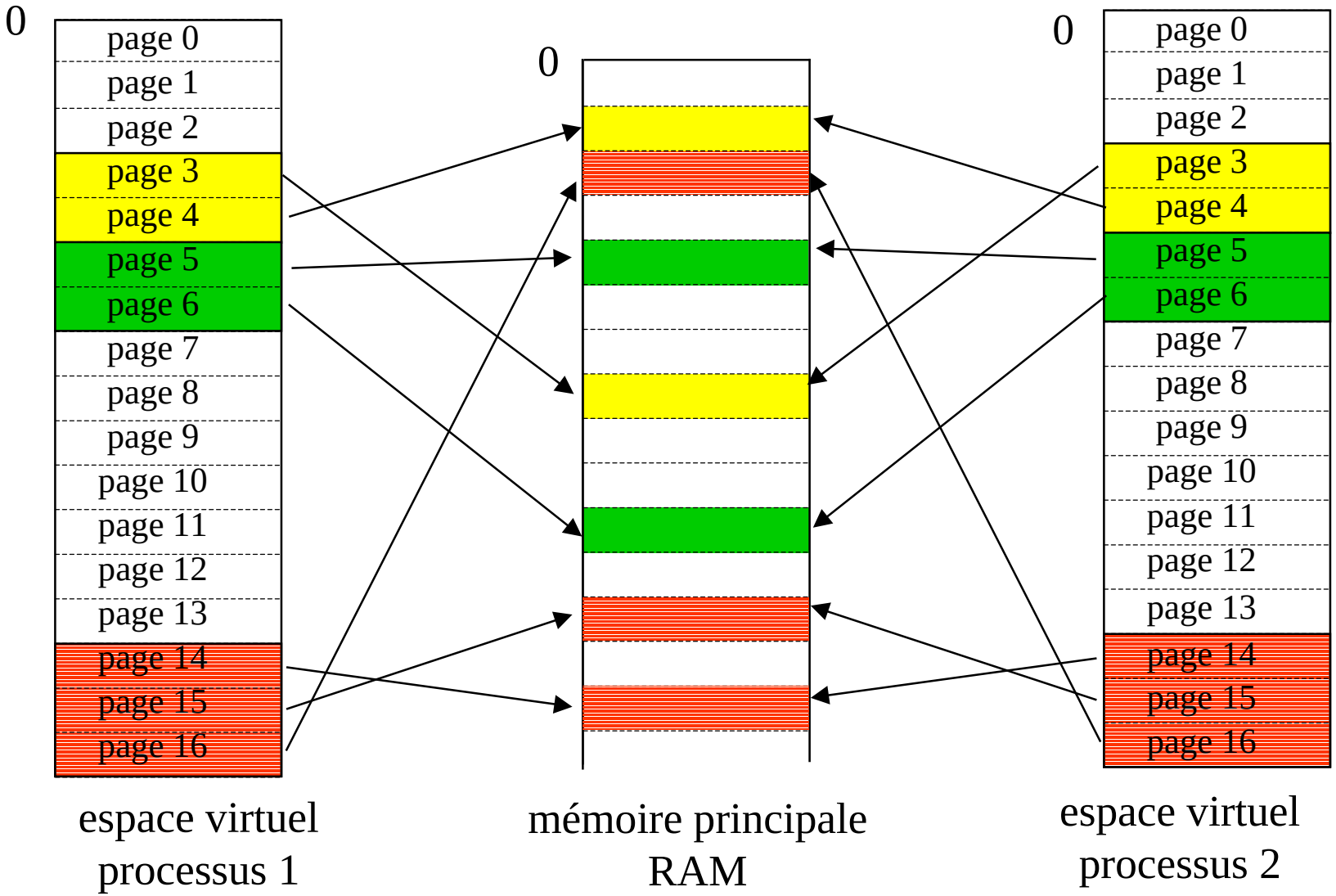
Partage de régions

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

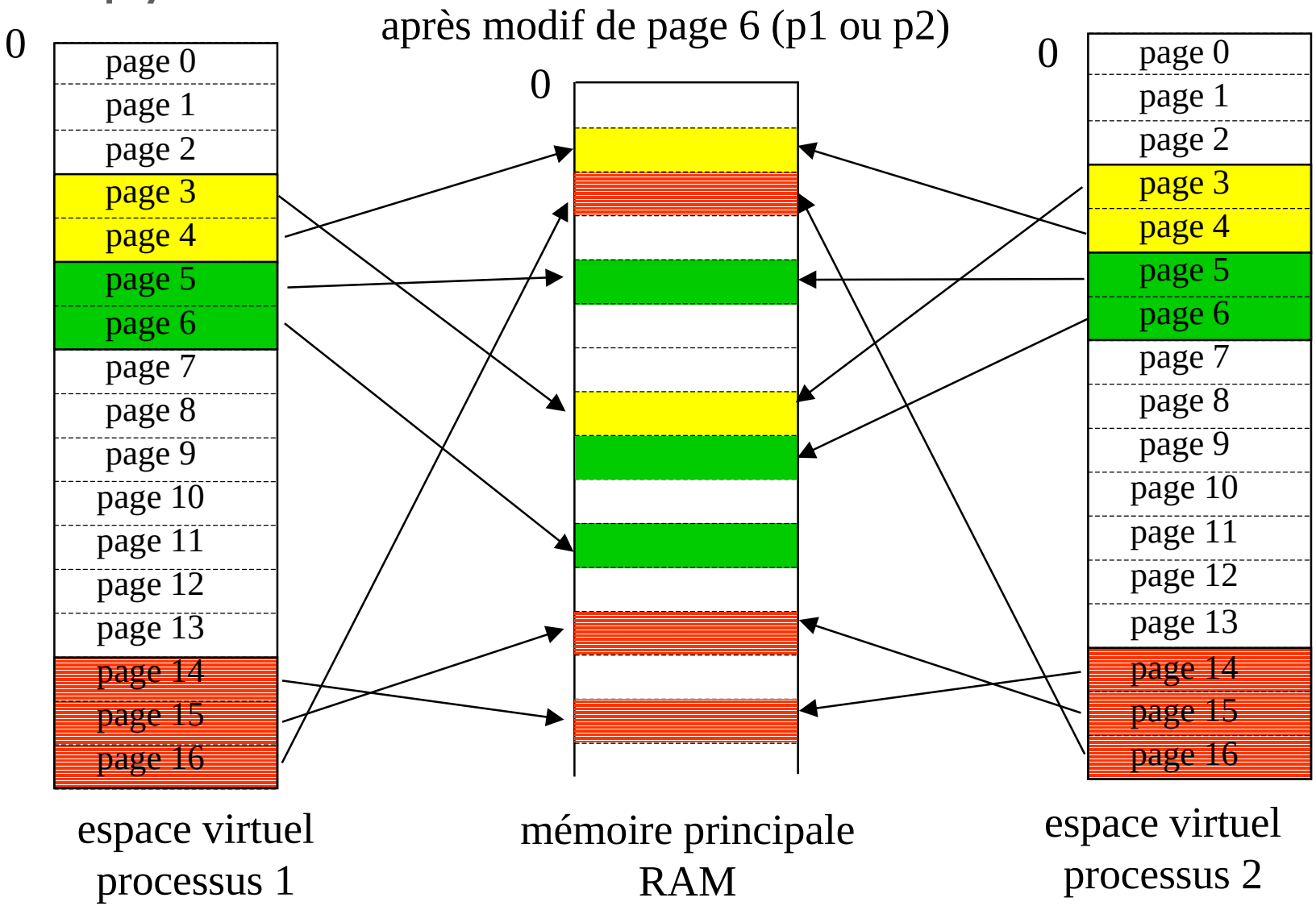
- partages de données
  - segments de données partagés
    - avantages :
    - *permet d'avoir des zones mémoire communes entre plusieurs processus, ce qui facilite les communication*
    - *communications rapides car ne passent pas par le noyau*
    - restriction :

    ne doit pas être utilisé pour transférer des pointeurs, car un pointeur est spécifique à un espace virtuel
    - inconvenient : la zone partagée n'est pas sécurisée

partage restreint :
copy-on-write



partage restreint :
copy-on-write



Pagination à la demande

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Le programme est paginé et les pages ne sont chargées en mémoire centrale qu'au moment ou elles sont nécessaires
- Les pages non chargées en mémoire centrale sont stockées sur la mémoire de réserve. Quand le programme veut utiliser une page non chargée, le système d'exploitation est activé par une trappe et il l'amène en mémoire centrale .

- dispositifs matériels nécessaires :
  - **un bit de présence par page dans la table des pages**

    BdP = **vrai** si la page est dans une case

    BdP = **faux** sinon.
  - un disque ou partie de disque pour la mémoire de réserve

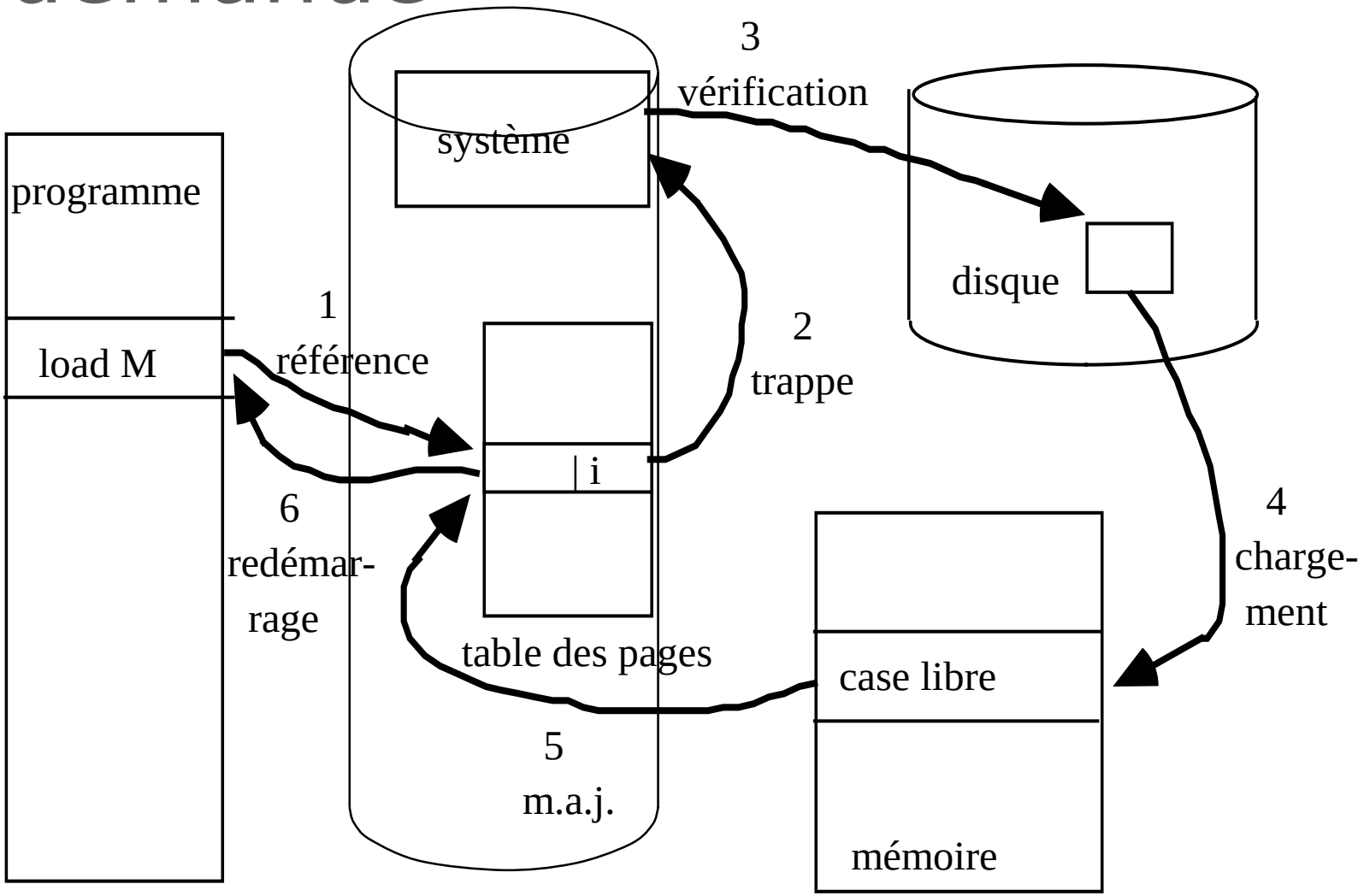
Pagination à la demande

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- calcul des adresses : identique à celui d'une mémoire paginée.
- Si la page est présente en mémoire centrale (BdP = vrai) l'instruction est exécutée normalement sinon le MMU provoque une trappe de type faute de page.

- **Traitement de la faute de page :**
- -1 vérifier que l'accès à cette page par le processus est autorisé;
- -2 trouver une case libre ;
- -3 trouver la page sur le disque et demander le chargement; commuter le processeur sur un autre processus;
- -4 attendre la fin du chargement de la page manquante;
- -5 modifier la table des pages;
- -6 attendre que le processus soit redevenu actif et le faire repartir sur la même instruction;

Pagination à la demande



Pagination à la demande

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- **Temps d'accès effectif**
- te = temps d'accès effectif;
- t_m = temps d'accès mémoire (environ 200 nanosec);
- p = probabilité de faute de page
- tfp = temps de traitement d'une faute de page (environ 8 msec);
- te= (1- p) * t_m + p * tfp
- te= (1- p) * 0,2 µsec + p * 8 000 µsec
- si p = 1/ 1000 alors te = 0,1999 + 8 = 8,2 µsec
-
- Pour avoir au plus 10 % de dégradation de te par rapport à t_m il faut p = $0{,}0000025 = 2 \times 10^{-6}$
- **Problème :**

  comment garder le taux de faute de page le plus faible possible ?

Pagination à la demande

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Comment obtenir un faible taux de FdP
  - constat de localité
    - localité spatiale : l'observation des programmes montre qu'ils sont relativement stables dans les pages utilisées (instructions proches, données proches)
    - localité temporelle : retour périodique vers des instructions (boucles)
    - avec une page de 4Koctets et des instructions ou des variables de 4octets on fait une faute de page pour le premier accès, mais pas pour les autres, donc $p < 0{,}001$

Pagination à la demande

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Comment obtenir un faible taux de FdP
- insérer le schéma de localisation des fautes de pages de Kaiser

politiques de remplacement de page

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Lorsque toutes les cases sont occupées il faut en libérer une avant de pouvoir l'utiliser

  (après recopie préalable sur disque si elle à été modifiée)

  - choix judicieux des victimes
    - politique FIFO : *on choisit en priorité les pages chargées le plus anciennement*
    - politique LRU (Least Recently used) : *on choisit en priorité les pages les moins récemment utilisées*
      - difficulté : il faudrait dater la dernière utilisation, mais ce doit être fait pas le matériel et en général il ne sait pas le faire

politiques de remplacement de page

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- choix judicieux des victimes
  - approximation de LRU : *il y a dans les TLB et les tables de pages un bit de référence qui est mis à 1 chaque fois qu'une adresse dans la page est recherchée.*
    - il suffit de remettre périodiquement ces bits à 0 pour voir qu'elles sont les pages qui n'ont pas été référencées pendant la dernière période

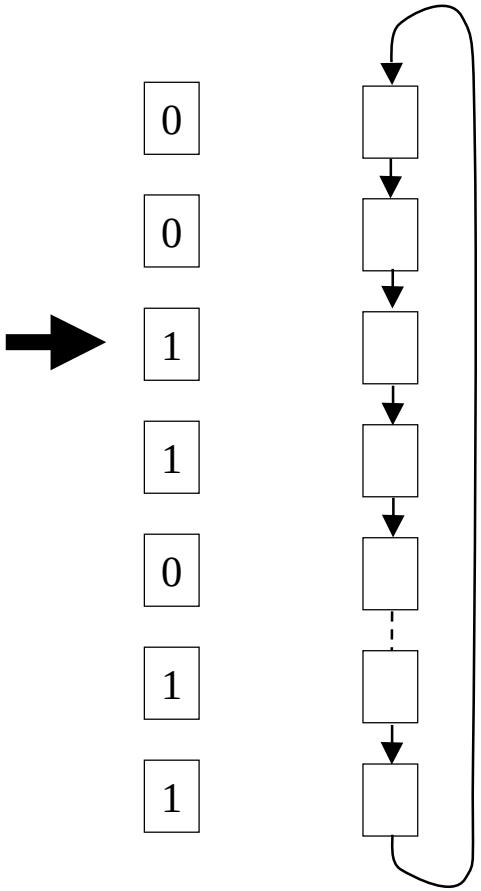
politiques de remplacement de page

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Algorithme de la seconde chance
  - chaque fois qu'une page est référencée son bit de référence est mis à 1.
  - les case victimes sont choisies selon la politique FIFO
  - Lorsqu'une page est choisie, on regarde son bit de référence
    - s'il vaut 0 la page est déchargée
    - s'il vaut 1 il est remis à 0 et la page n'est pas déchargée (obtient une seconde chance de rester en mémoire)
  - implémentation possible avec une liste circulaire des pages

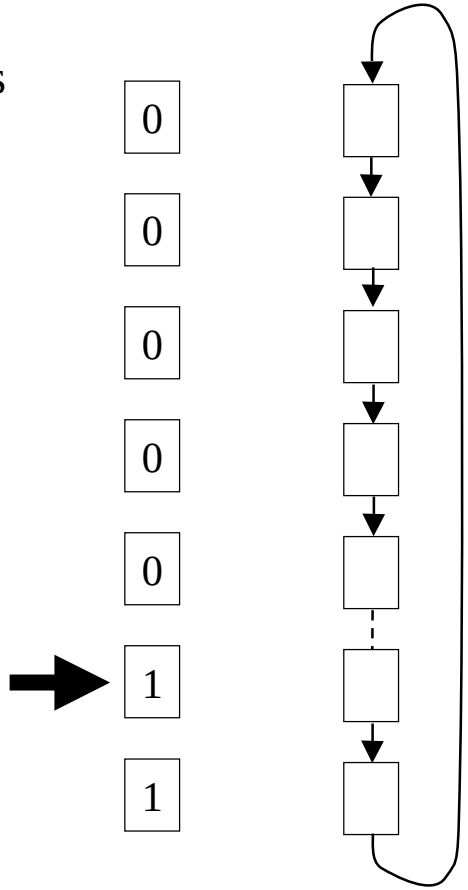
politiques de remplacement de page

- Algorithme de la seconde chance



les pages sont chaînées circulairement et examinées selon ce chaînage

si le bit vaut 1 on le remet à 0 et on passe à la suivante



politiques de remplacement de page

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- **Le voleur de page**
  - Il est réveillé quand le nombre de pages libres tombe en dessous d'un seuil B et se suspend lorsque le nombre de cases libres atteint un seuil H
  - Pour chaque case de la mémoire
  - Si la case a son bit de référence à 0 alors le voleur incrémente une variable "age" associée à la case, sinon le bit de référence et l'age sont remis à zéro.
  - lorsque l'age dépasse une valeur donnée la case est placée dans l'état disponible pour le déchargement (elle ne sera déchargée effectivement qu'en cas de besoin de place, on lui laisse ainsi la possibilité d'être réutilisée si elle est à nouveau référencée).

politiques de remplacement de page

- **Le voleur de page** ($$refaire le schéma)

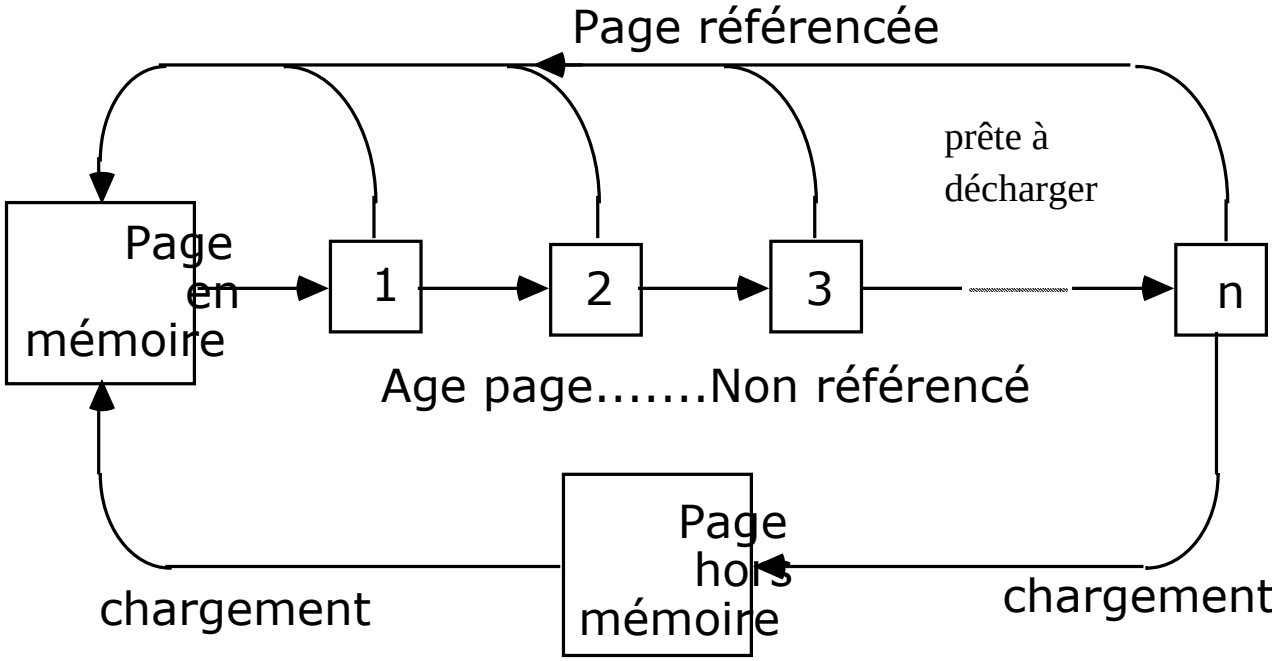


Schéma de vieillissement d'une page

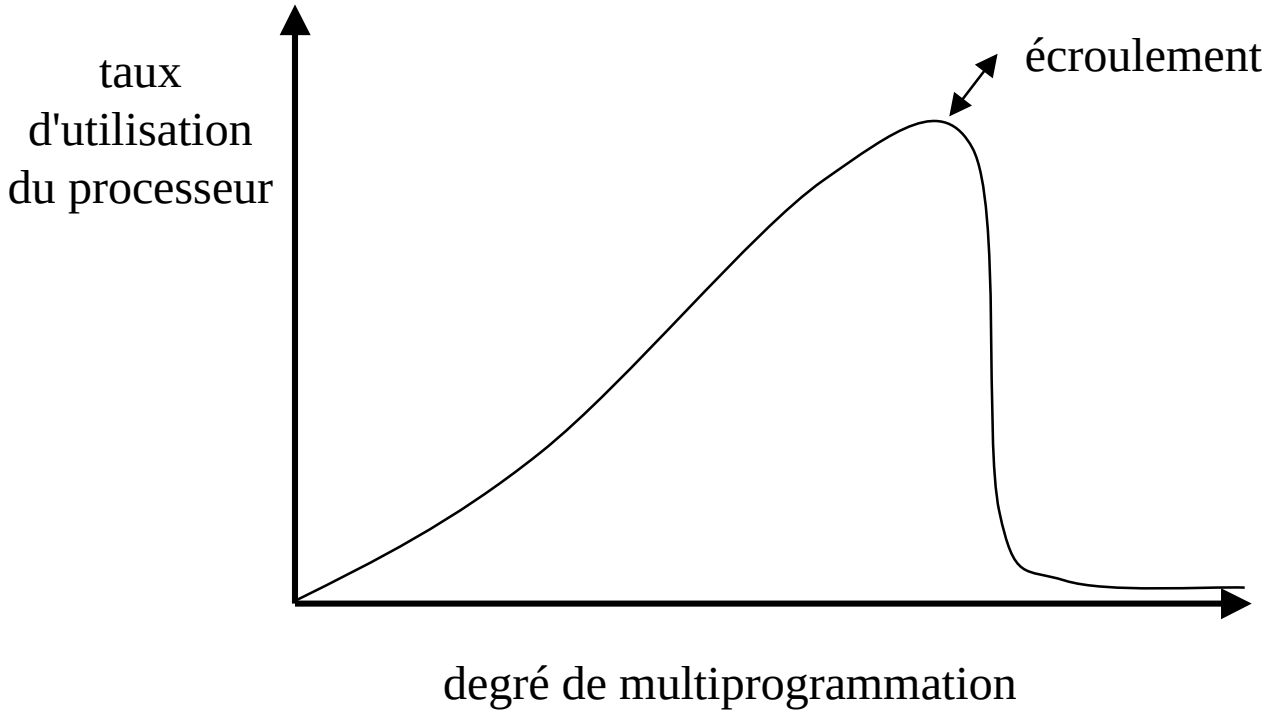
politiques de remplacement de page

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Allocation de cases
  - remplacement local / remplacement global
  - il faut donner un minimum de cases à chaque processus, sinon il risque de faire des fautes de pages fréquemment
  - mais si on lui en donne trop il y aura des cases non utilisées
    - allocations égales pour tous les processus
    - allocation proportionnelle à la taille de l'image mémoire
  - il faut éviter qu'un afflux de travaux ne provoque un écroulement : le nombre de pages allouées à chacun d'eux étant devenu trop faible, il font des FdP très fréquemment, d'où attente de pages mais n'avancent pas

politiques de remplacement de page

- Ecroulement (Thrashing)



politiques de remplacement de page

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Working Set
  - d'après le constat de localité, un processus n'accède pas en permanence à la totalité de son espace virtuel mais se cantonne à des parties limitées pendant un certain temps puis change ensuite. On appelle une telle partie un working set.
  - on essaye de donner à un processsus à chaque instant assez de cases pour mettre le working set de l'instant
  - les fautes de pages ne se produiront que lors des changements de working sets
  - Difficulé : estimer et prvoir correctement la taille du working set à chaque moment

politiques de remplacement de page

ensiie
école nationale supérieure d'informatique pour l'industrie et l'entreprise

- Working Set
  - $$inserer le schema d'évolution du nb de FdP de Silberschatz