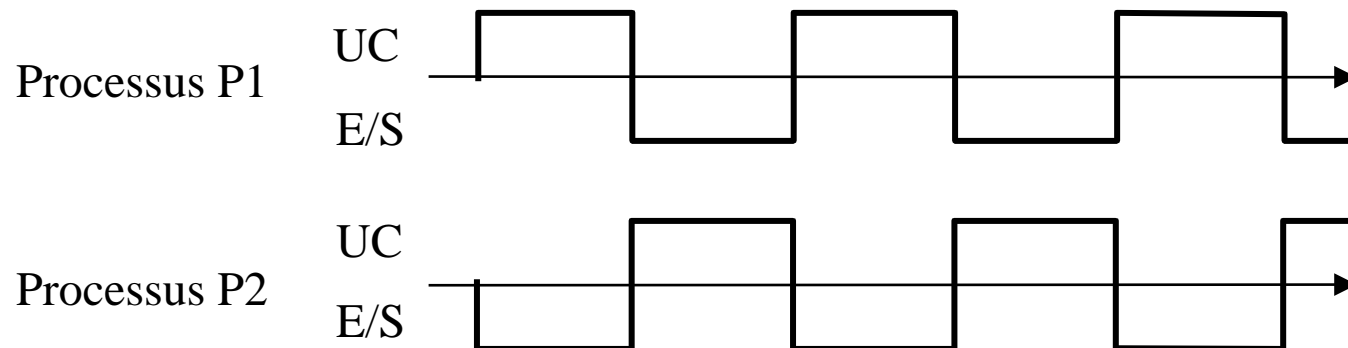


# La gestion de la mémoire centrale

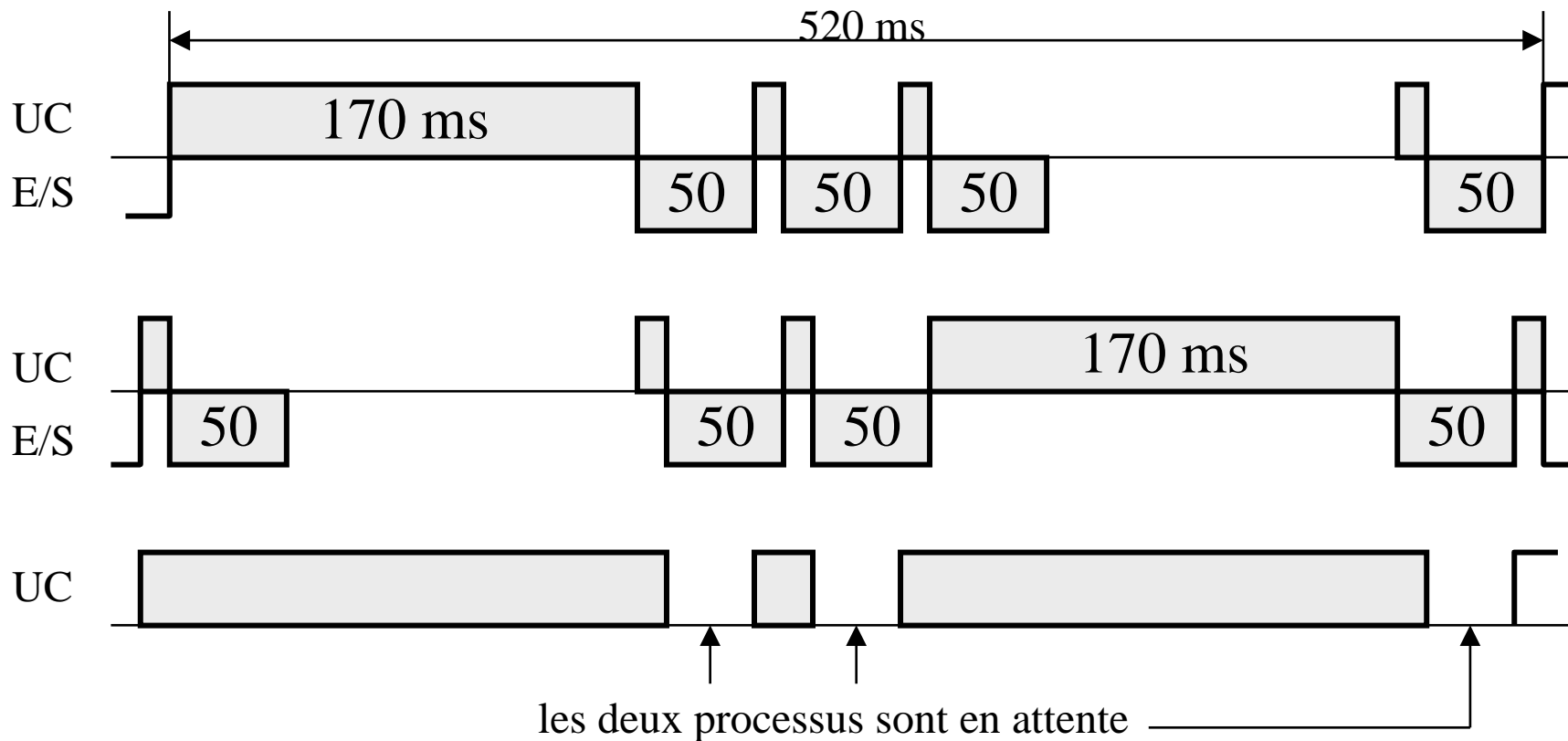
# Notion de multiprogrammation (1)

- Un seul processus en mémoire, en attente de ressource  
le processeur ne fait rien
- Plusieurs processus en mémoire, partage du processeur  
ci-dessous, le processeur travaille à 100%  
situation idéale, en fait pas tout à fait aussi bien



# Notion de multiprogrammation (2)

- Activité processeur: 400 ms sur 520 ms, soit 77%



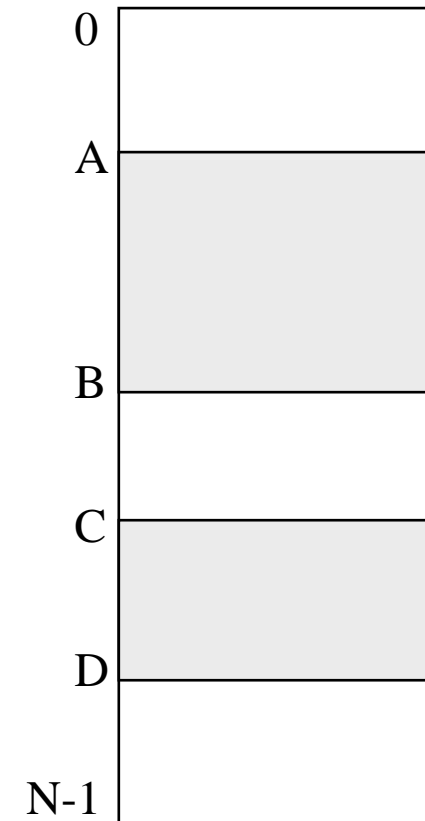


# Problèmes à résoudre

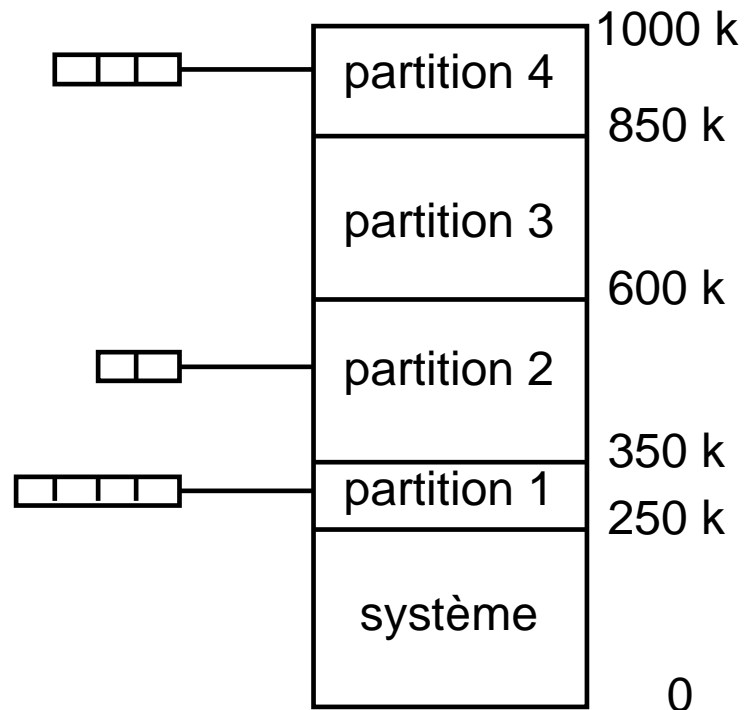
- Définition d'un espace propre à chaque processus
    - indépendance des processus du point de vue désignation des objets
  - Protection des processus entre eux
    - indépendance du point de vue des actions
  - Attribution d'un espace de mémoire physique à chacun
    - indépendance du point de vue de la localisation
- 
- ```
graph TD; A[● Définition d'un espace propre à chaque processus] --> B[● Protection des processus entre eux]; B --> C[● Attribution d'un espace de mémoire physique à chacun];
```
- traduction*

# Le partitionnement (1)

- Mémoire centrale
  - emplacements  $0..N-1$ ,
  - découpée en espaces disjoints  $\Rightarrow$  partition
- Espace propre =  $[A..B]$ 
  - $[A..B]$   $[C..D] =$  , i.e.  $B < C$   $D < A$
- Protection
  - adresses autorisées dans  $[A..B]$
- Espace mémoire physique =  $[A..B]$

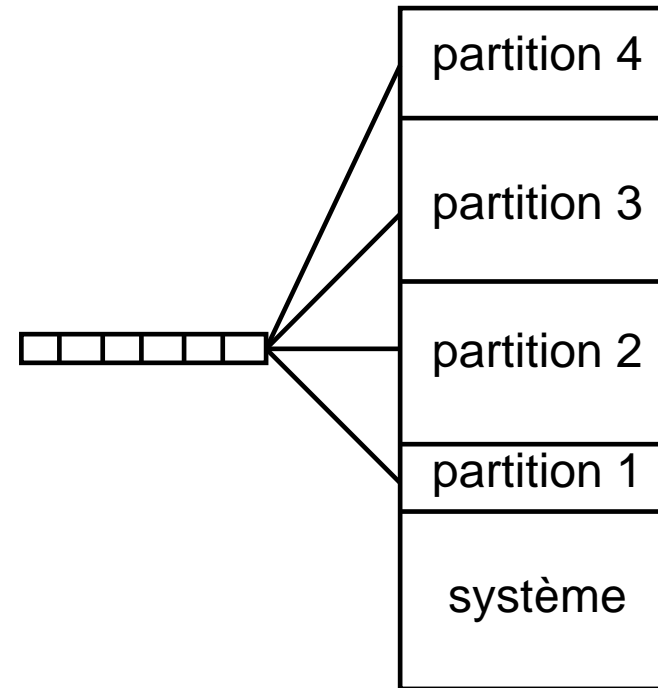


# Le partitionnement (2)



files multiples

partition fixée à l'édition de liens  
=> 3 est vide



file unique

choix de partition au chargement  
=> translation

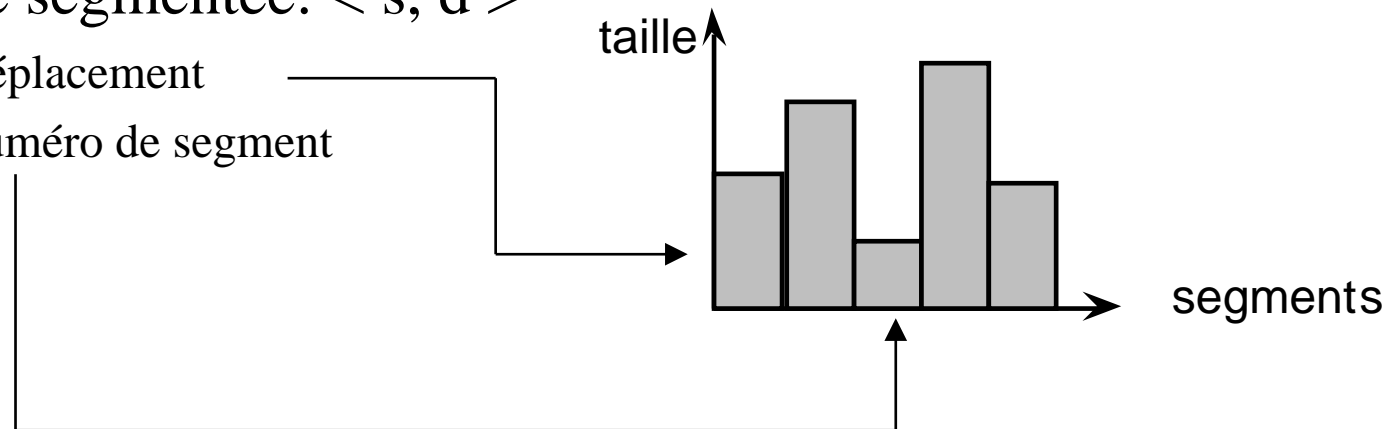
*inconvénient: mauvaise utilisation car les processus obtiennent plus que nécessaire*

# Notion de mémoire segmentée

- Faciliter la gestion par les processus de leur espace propre
  - espace à deux dimensions
  - pris en compte par le matériel sur Multics, iAPX286, iAPX386...
  - simulé par éditeur de liens et le chargeur
- Adresse segmentée:  $\langle s, d \rangle$

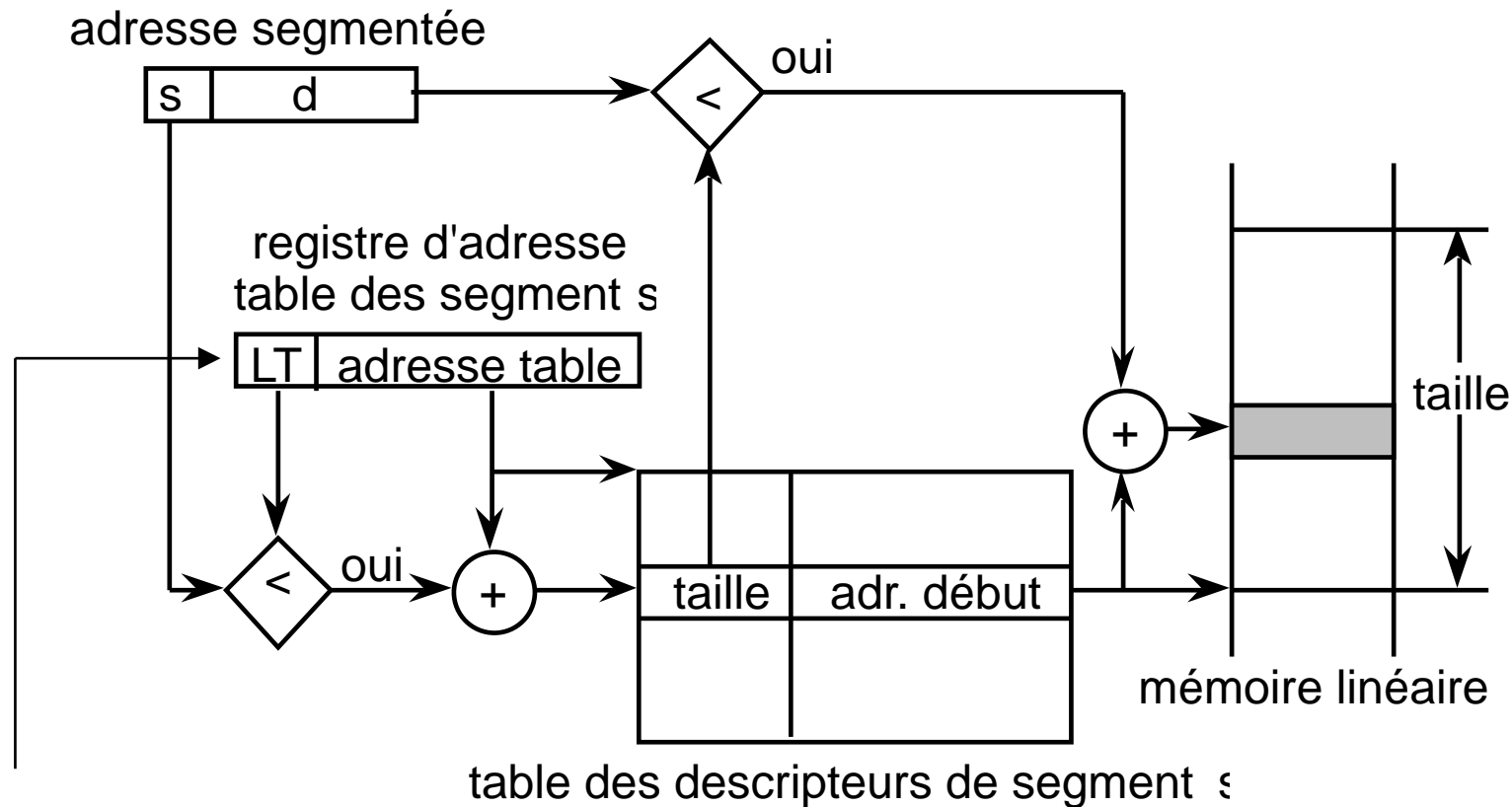
d = déplacement

s = numéro de segment





# Adressage segmenté intel

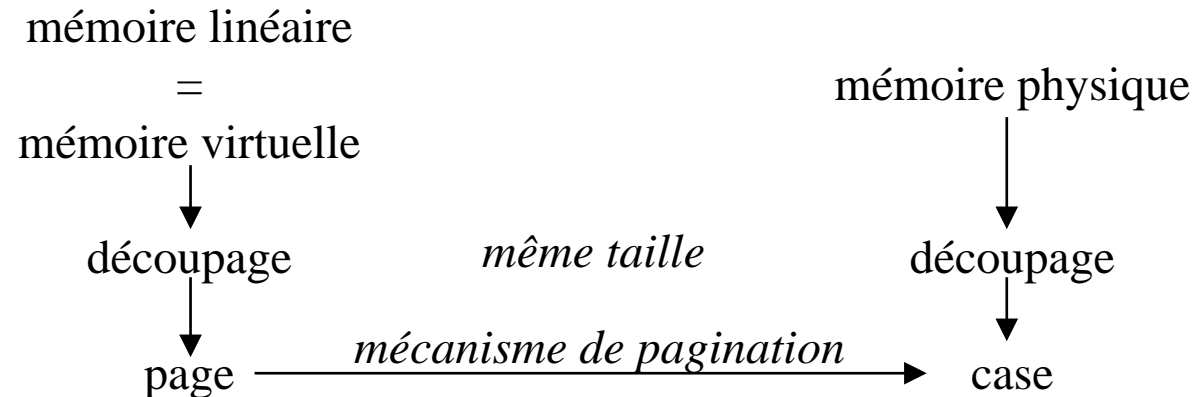


*dans un registre processeur  
lorsque le processus dispose  
du processeur*

*propre au processus*

# Mécanisme de pagination

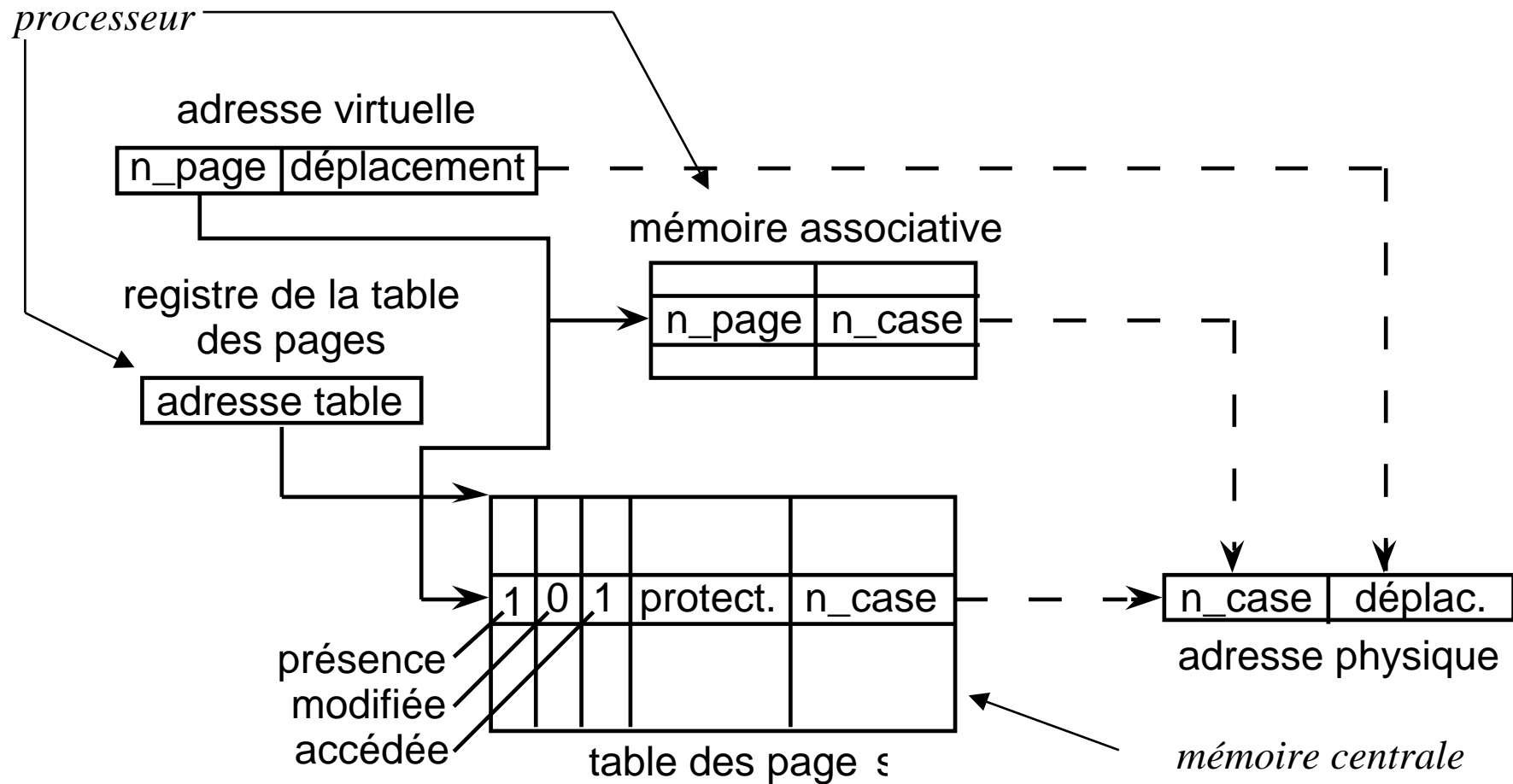
- Lever la contrainte de contiguïté de l'espace de mémoire physique alloué à un processus



*remarques:*

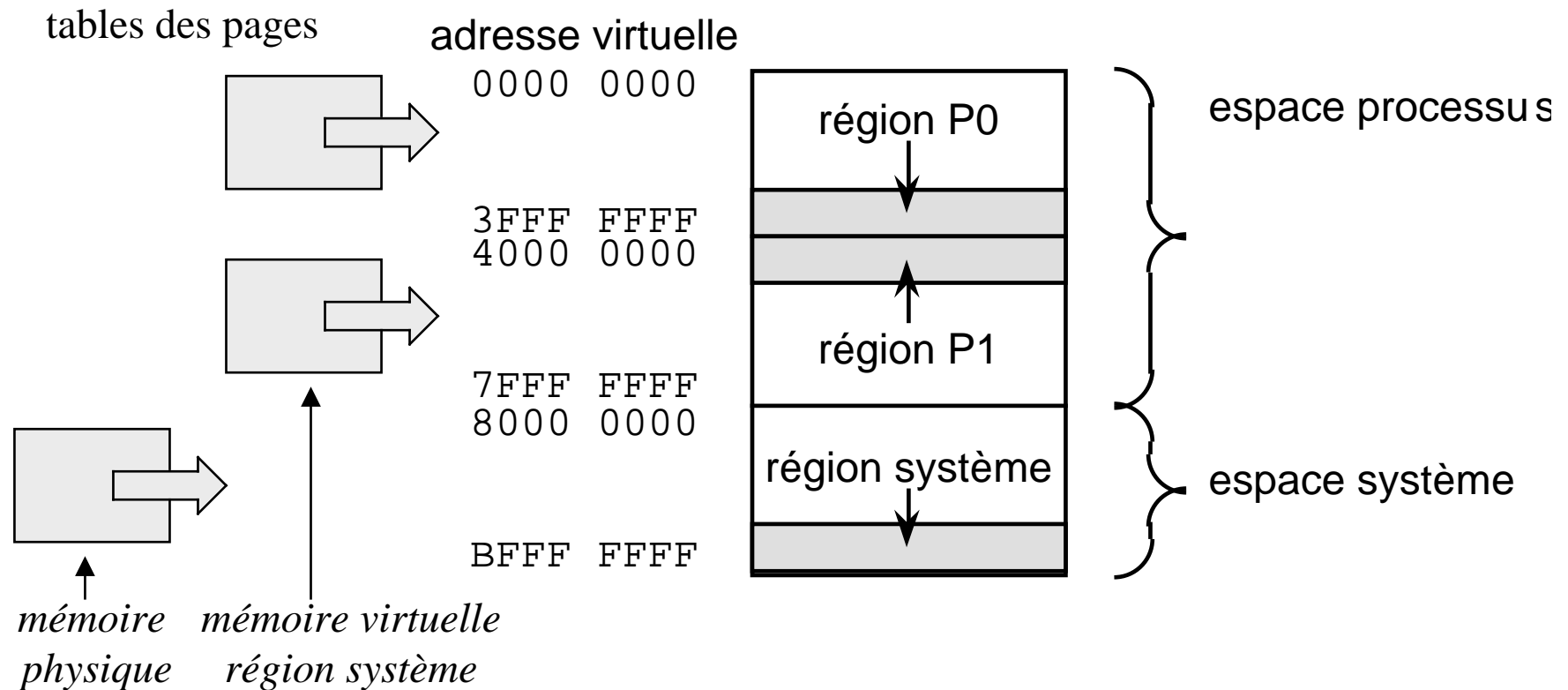
*support matériel obligatoire, car traduction à chaque accès*  
*allocation peut être dynamique*  
*toutes les pages n'ont pas forcément une case allouée*

# Pagination à un niveau

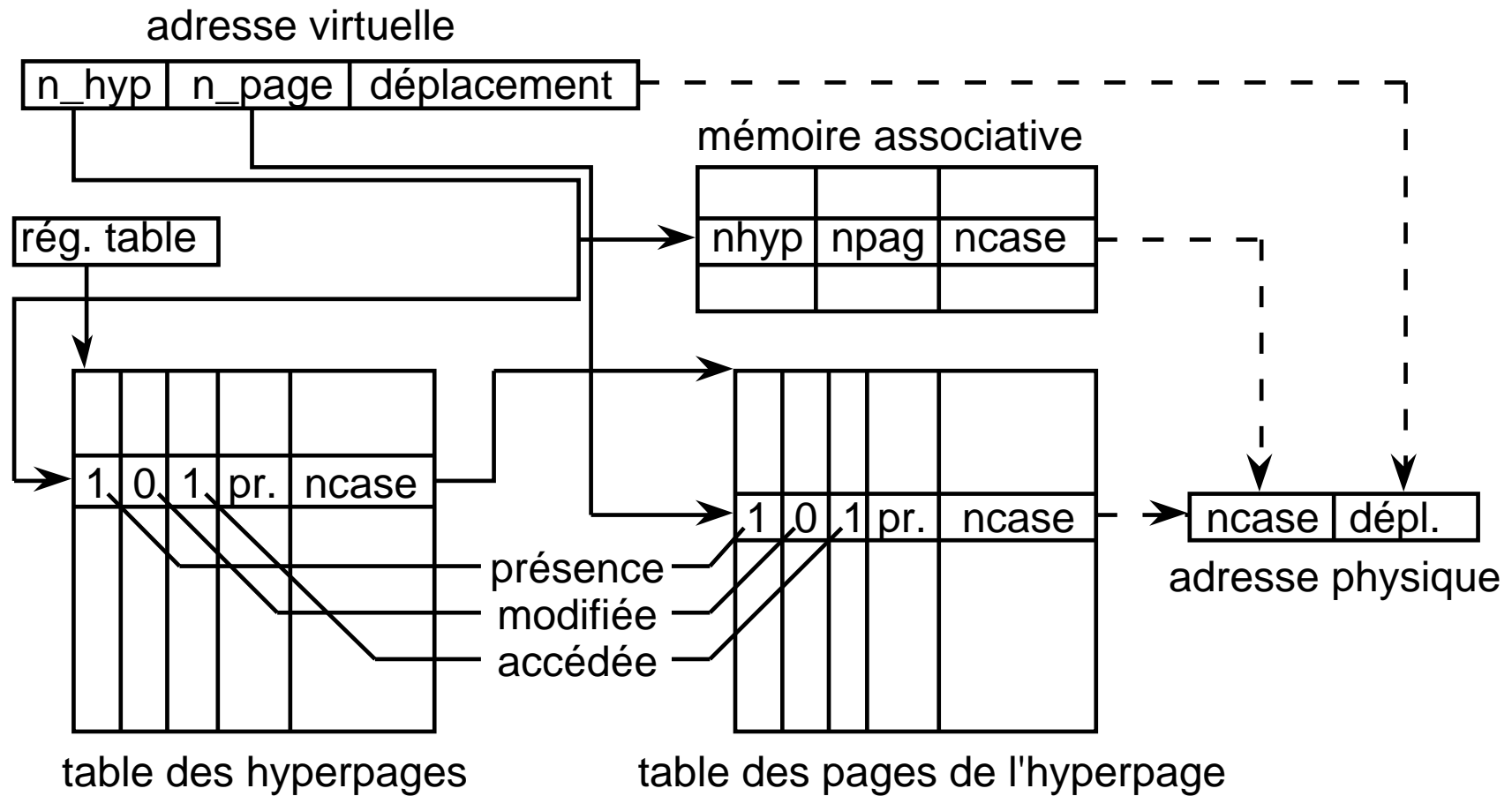


# Pagination sur VAX-VMS

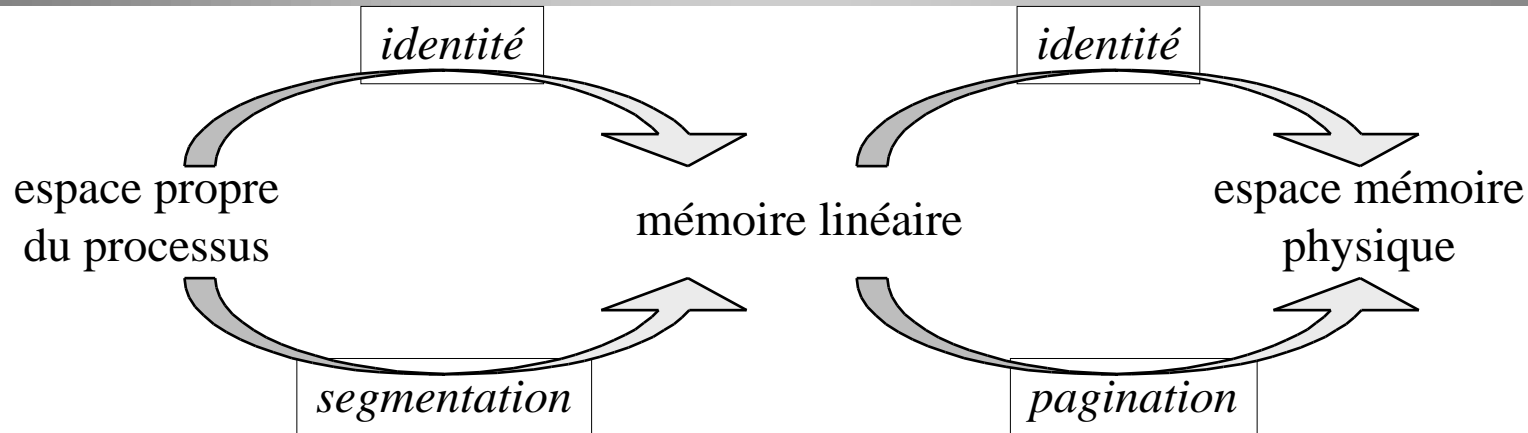
Pages de 512 octets, 3 régions, 3 tables de longueur variable, 3 registres



# Pagination à deux niveaux



# Segmentation et pagination (1)

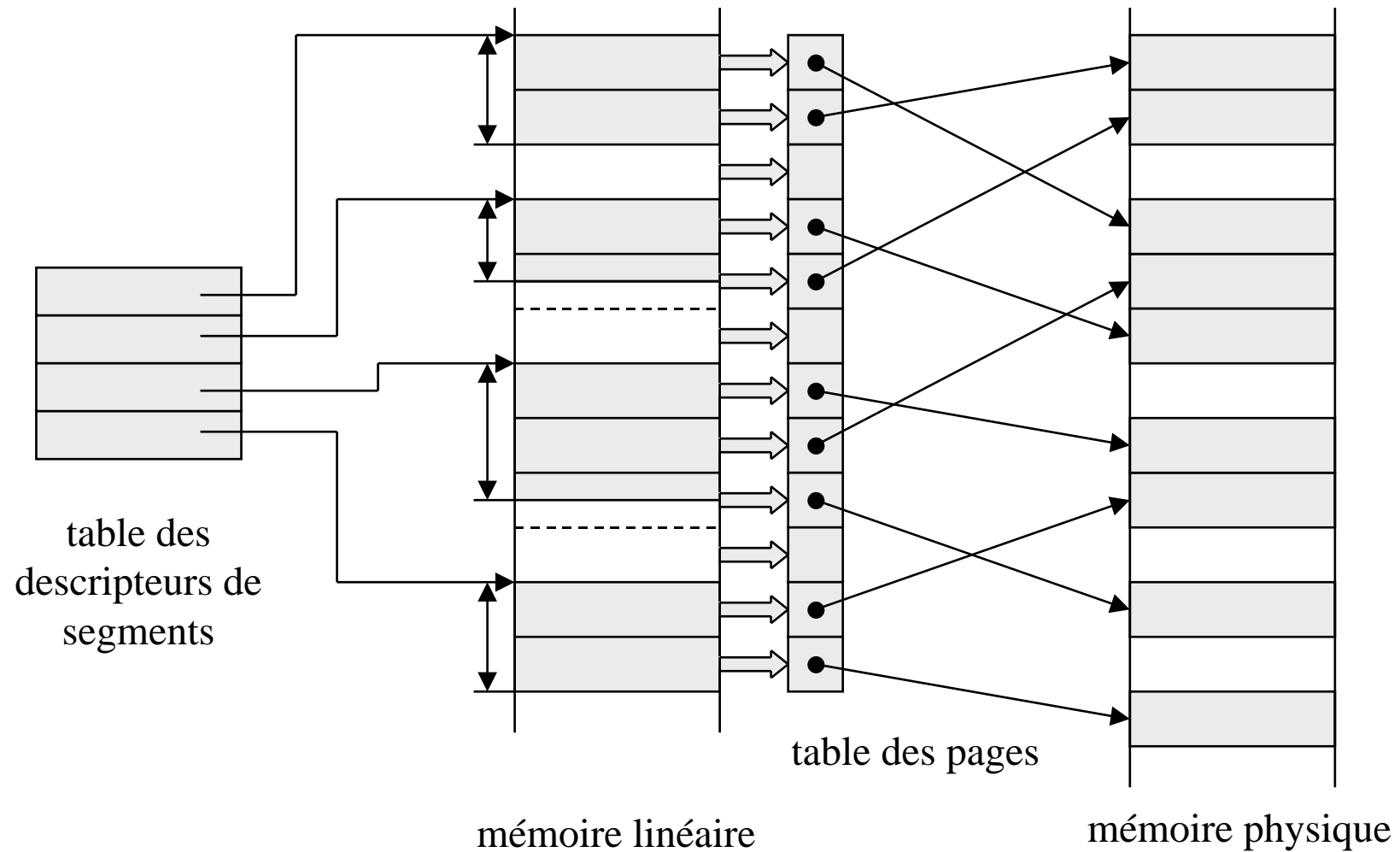


2 dimensions  
allocation par zone en mémoire linéaire  
déplacement pour éviter les trous

allocation par page en mémoire physique  
pas de contiguïté  
trous sans perte  
*pages libres => pas de cases alloués*  
*déplacement => changement de table des pages*

*partage de données entre processus => portions de mémoire physique communes*  
*pages associées à une même case*  
*partage de la table des pages d'une hyperpage*  
*sans pagination, même zone de mémoire allouée à deux segments*

# Segmentation et pagination (2)



# Pagination à la demande

- Augmenter le degré de multiprogrammation, en donnant moins de mémoire à chacun
- Donner l'illusion d'une mémoire plus grande que la réalité
- À chaque page, bit de présence:
  - 1 = page allouée => accès
  - 0 = *défaut de page*, et déroutement système
    - s'il y a une case libre => allouer la case
    - s'il n'y en a pas => remplacer une page présente
      - *choisir une case occupée*
      - sauvegarder éventuellement son contenu
      - charger la page depuis le disque dans la case
    - modifier les tables puis accès



# Algorithmes de remplacement (1)

- Optimal: celle qui sera accédée dans un avenir le + lointain  
le meilleur, mais hélas non réalisable
- Fréquence d'utilisation (LFU) ← *matériel spécifique*
- Chronologie d'utilisation (LRU) ← *matériel spécifique*
- Chronologie de chargement (FIFO) ← *purement logiciel*
- Aléatoire (Random) ← *purement logiciel*
- Seconde chance  
comme FIFO, sauf si *accédée* = 1  
0 et tête

# Algorithmes de remplacement (2)

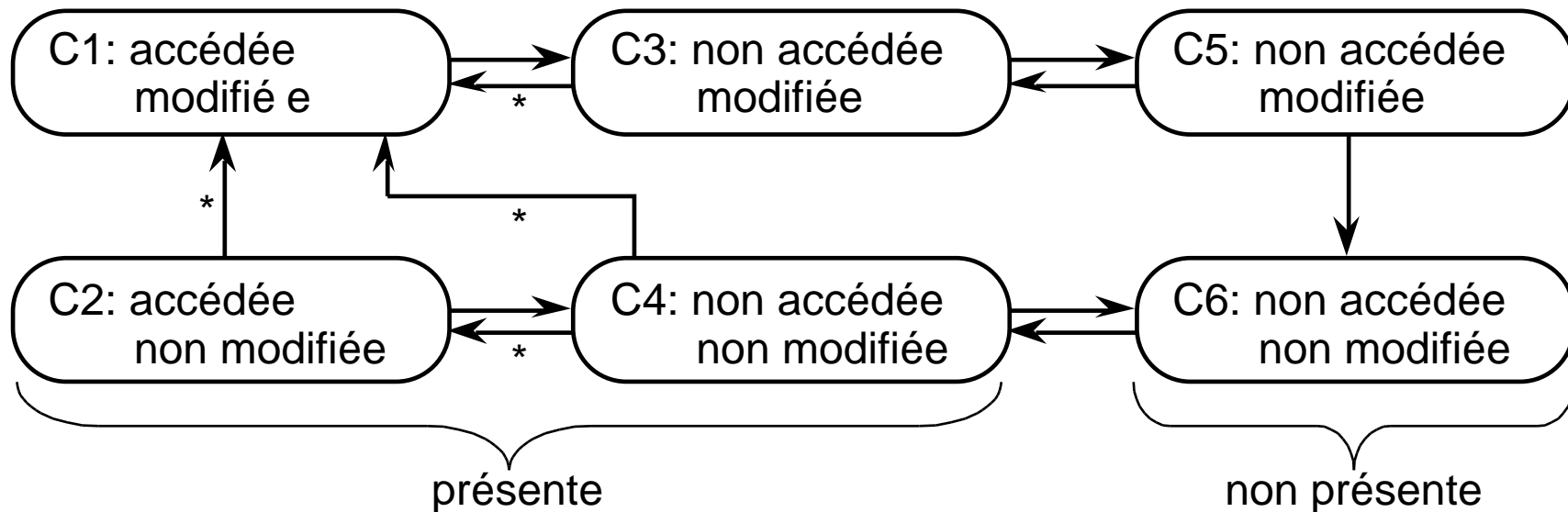
- Non récente utilisation (approcher LRU sans matériel)

bits accédée, modifiée et présence

6 catégories => classement des pages

transition entre catégories par matériel (\*) ou

par logiciel (périodique): présente := accédée; accédée := 0



# Notion d'espace de travail

- Toutes les pages nécessaires sont en mémoire => 0 défauts
- Processus à l'étroit => remplacements de pages fréquents
- Remplacement de page = attente de chargement (disque)

$$T_d = p \frac{T}{t} T_{uc}$$

*probabilité de défaut de page* →  $p$   
*temps total attente de page* →  $T_d$   
*temps attente une page* →  $T$   
*temps total UC* →  $T_{uc}$   
*temps accès mémoire centrale* →  $t$

- Problème:  $T \gg 10000 t$  => effet brutal (*écroulement*)
- Espace de travail d'un processus  $T(P)$  = nombre de pages nécessaires

$$T(P) \leq M$$

*processus présents en mémoire* →  $P$   
 $E$   
*taille mémoire disponible* →  $M$

# Conclusion (1)

- Multiprogrammation => récupérer les temps d'attente d'un processus pour les autres
- 3 problèmes:
  - espace adresse de chacun
  - protection des processus entre eux
  - allocation de mémoire physique à chacun
- Partitionnement
  - découpage statique de la mémoire physique
  - mauvaise utilisation

# Conclusion (2)

- **Segmentation**
  - espace à deux dimensions pour chaque processus
  - structuration de l'espace d'un processus
- **Pagination**
  - plaquer la mémoire virtuelle sur la mémoire physique
  - découpage en pages et cases de même taille
  - les pages peuvent être mises dans n'importe quelle case
  - pagination à 1 niveau => taille de la table des pages
  - pagination à 2 niveaux
    - table des hyperpages
    - tables des pages des hyperpages
- **Utilisation conjointe de la segmentation et de la pagination**

# Conclusion (3)

- **Pagination à la demande**
  - n'allouer une case que si le processus a besoin de la page
  - algorithmes de remplacements logiciels ou avec aide du matériel
- **Espace de travail**
  - pages nécessaires au processus à un instant donné
  - déterminer le nombre de cases correspondant
  - s'il en a plus => mauvaise utilisation de la mémoire
  - s'il en a moins => ralentissement et risque d'écroulement du système