Gestió de la memòria. Motivació

- Com organitzem la memòria de forma que els programes d'usuari i el sistema operatiu la puguin compartir?

Gestió de la memòria. Motivació

- Com organitzem la memòria de forma que els programes d'usuari i el sistema operatiu la puguin compartir?
- Com es protegeixen els diferents àmbits?

Gestió de la memòria. Motivació

- Com organitzem la memòria de forma que els programes d'usuari i el sistema operatiu la puguin compartir?
- Com es protegeixen els diferents àmbits?
- Com vinculem les adreces amb el programa un cop sabem on han de residir?

Gestió de la memòria. Motivació

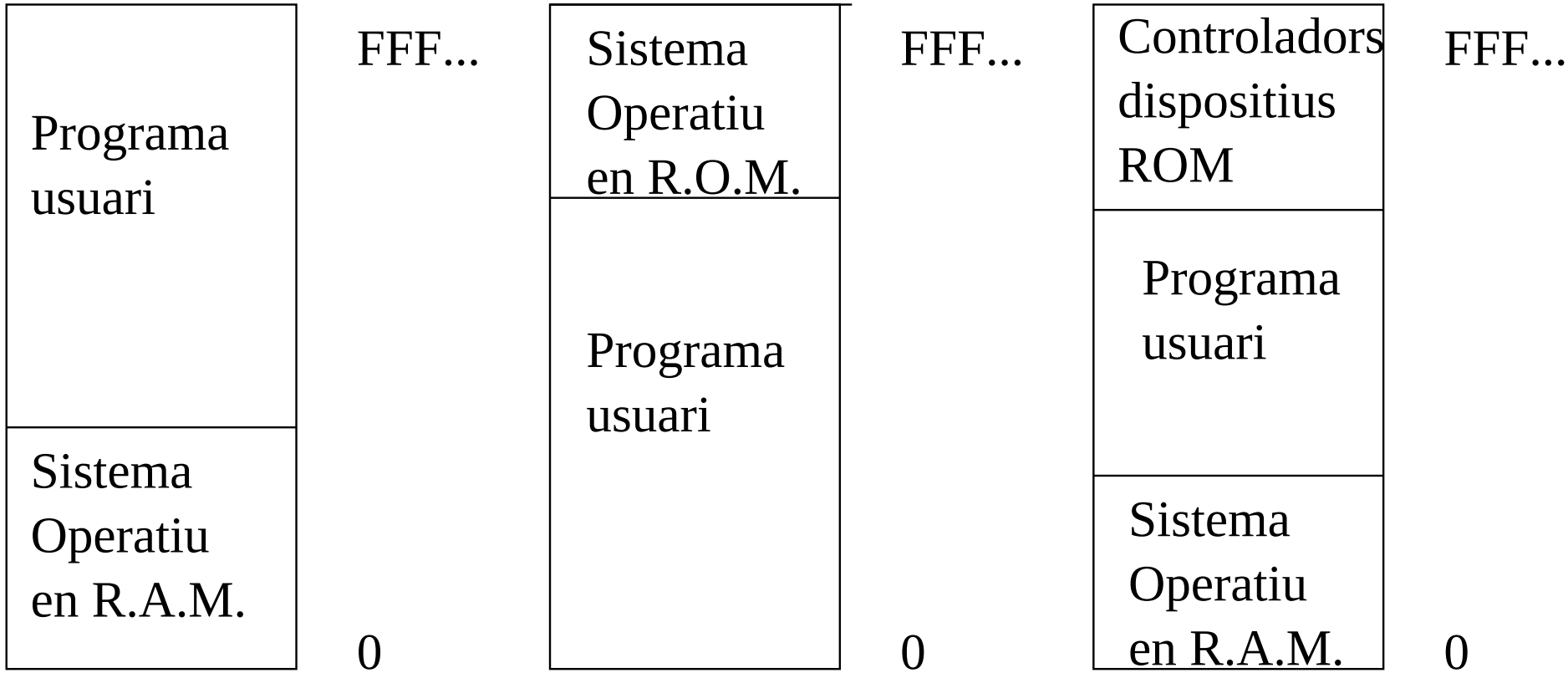
- Com organitzem la memòria de forma que els programes d'usuari i el sistema operatiu la puguin compartir?
- Com es protegeixen els diferents àmbits?
- Com vinculem les adreces amb el programa un cop sabem on han de residir?
- Què fem si tot no cap a la memòria principal?

GIE/ICE

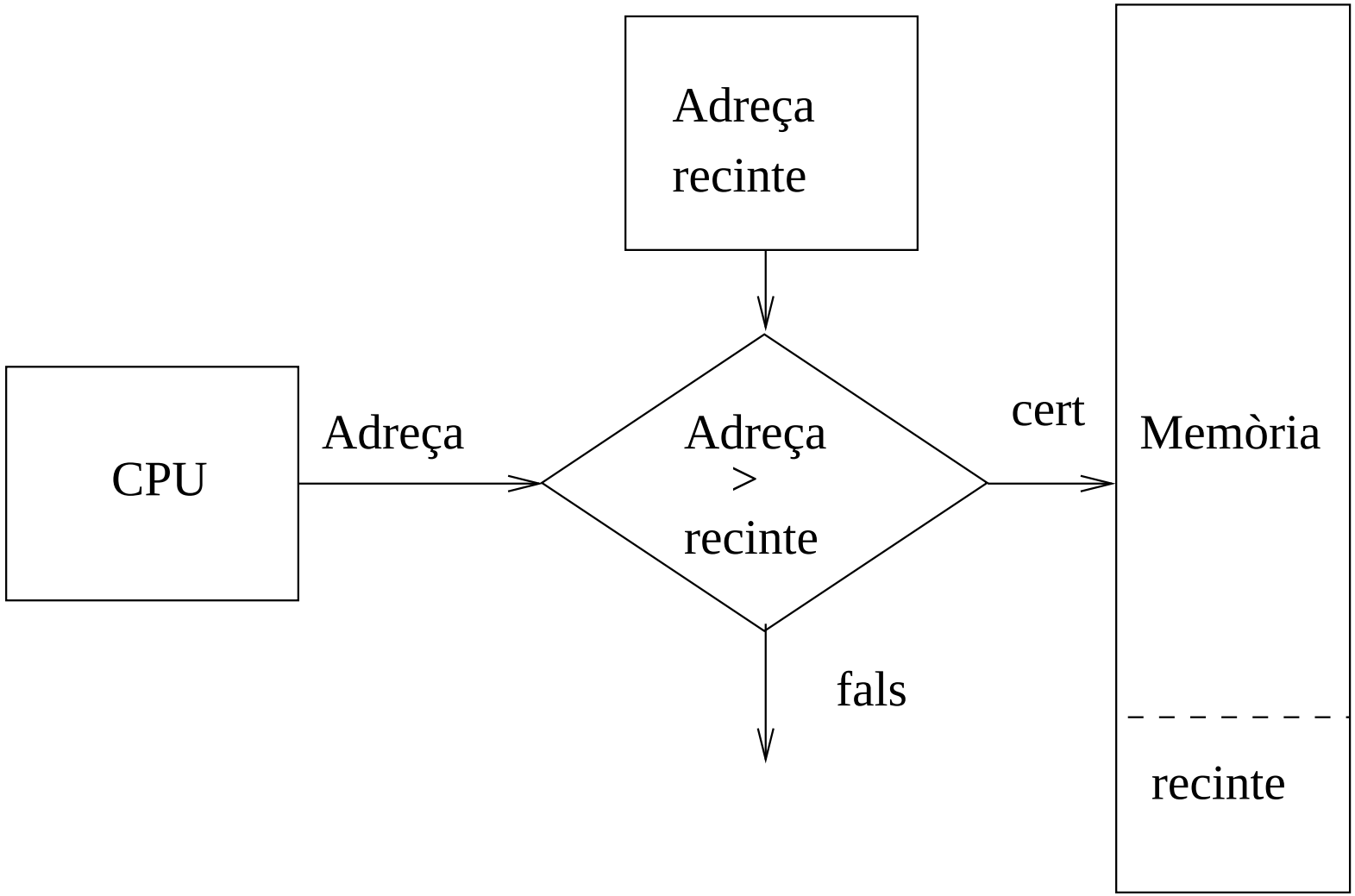
Gestió de la memòria. Motivació

- Com organitzem la memòria de forma que els programes d'usuari i el sistema operatiu la puguin compartir?
- Com es protegeixen els diferents àmbits?
- Com vinculem les adreces amb el programa un cop sabem on han de residir?
- Què fem si tot no cap a la memòria principal?
- Què fem si un programa és més llarg que no cap a la memòria principal?

Monoprogramació (I)



Protecció de hardware mitjançant un registre de recinte.



Monoprogramació (III). Ubicació dinàmica

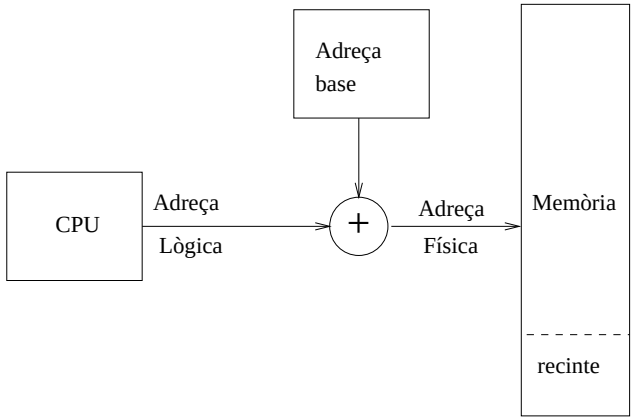
- La càrrega de programes d'usuari s'hauria de fer a partir de l'adreça del recinte.

Monoprogramació (III). Ubicació dinàmica

- La càrrega de programes d'usuari s'hauria de fer a partir de l'adreça del recinte.
- El compilador genera codi reubicable. El muntador i el servei que carrega el programa genera la ubicació definitiva.

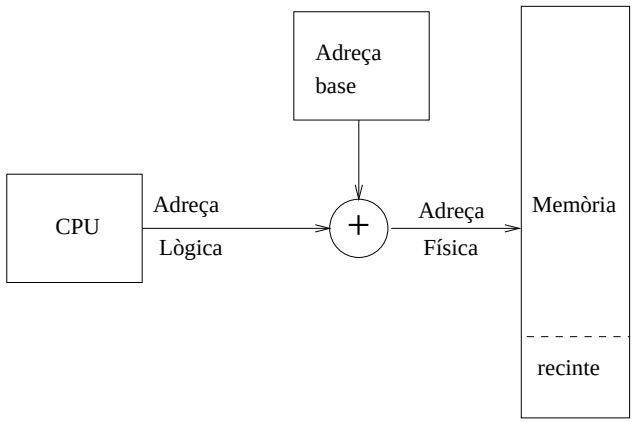
Monoprogramació (III). Ubicació dinàmica

- La càrrega de programes d'usuari s'hauria de fer a partir de l'adreça del recinte.
- El compilador genera codi reubicable. El muntador i el servei que carrega el programa genera la ubicació definitiva.
- En la execució d'un programa l'adreça de recinte no pot variar. Si el sistema operatiu vol afegir més codi per què així ho demana l'usuari, poden haver problemes. Una forma de solucionar-ho és mitjançant la **reubicació dinàmica**



Monoprogramació (III). Ubicació dinàmica

- La càrrega de programes d'usuari s'hauria de fer a partir de l'adreça del recinte.
- El compilador genera codi reubicable. El muntador i el servei que carrega el programa genera la ubicació definitiva.
- En la execució d'un programa l'adreça de recinte no pot variar. Si el sistema operatiu vol afegir més codi per què així ho demana l'usuari, poden haver problemes. Una forma de solucionar-ho és mitjançant la **reubicació dinàmica**



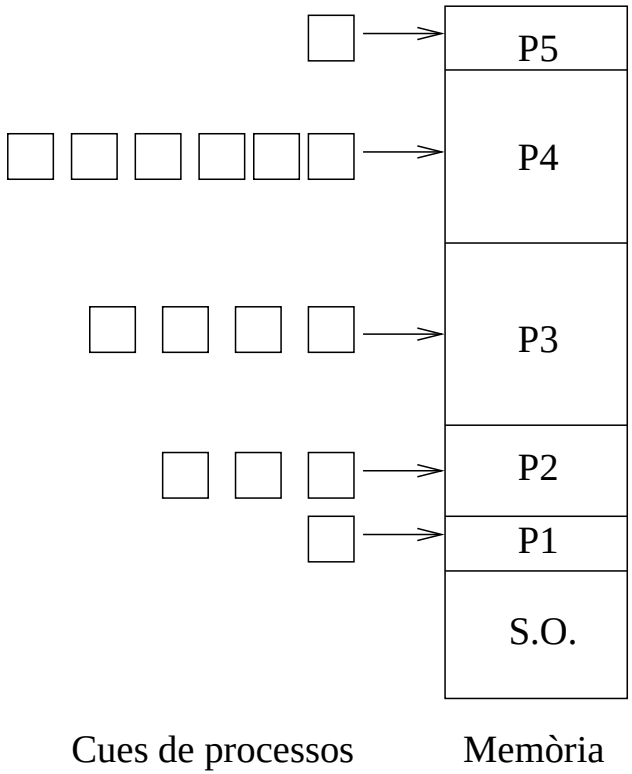
- Ara es pot canviar el registre en temps d'execució però cal moure el programa.

Multiprogramació. Particions fixes

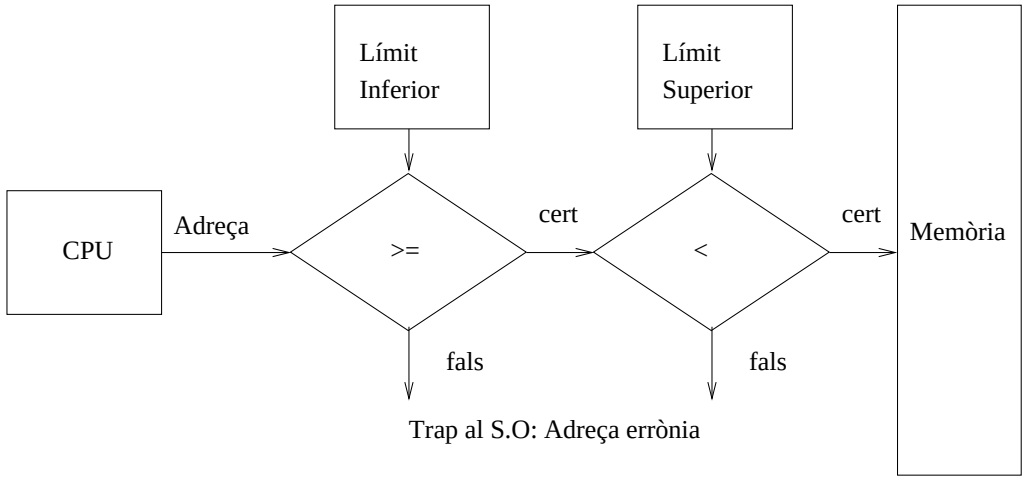
- Es divideix la memòria en *n* particions probablement desiguals.

Multiprogramació. Particions fixes

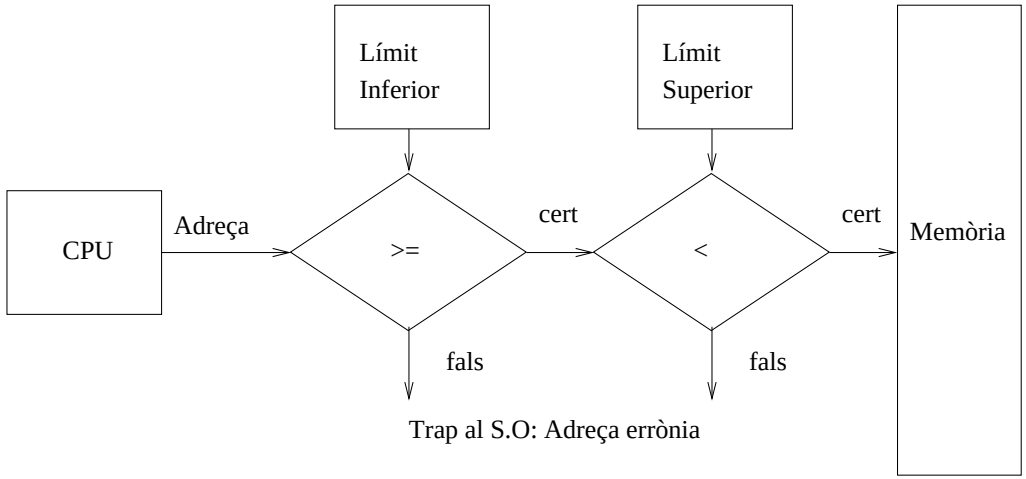
- Es divideix la memòria en *n* particions probablement desiguals.
- La planificació de processos haurà de tenir en compte si el procés cap en una partició donada.



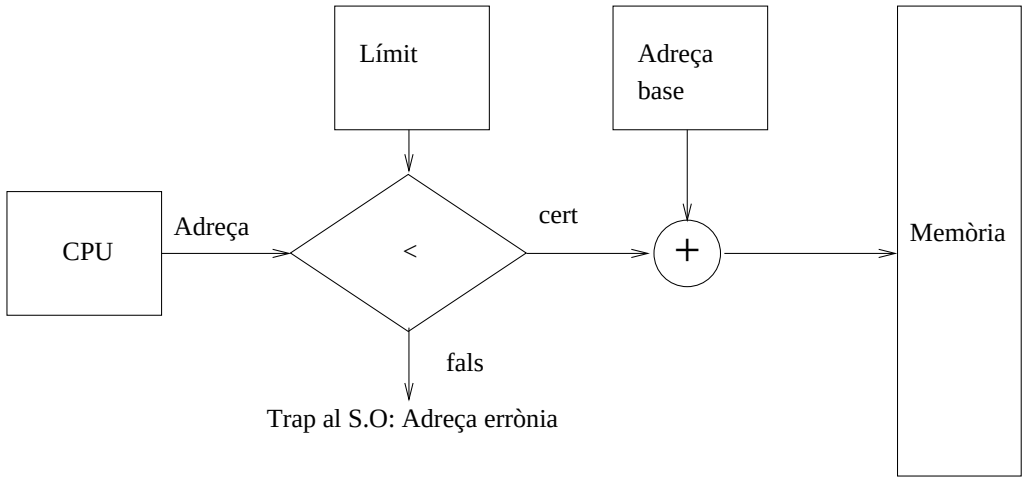
- Registres límits



- Registres límits



- Registres base/límit



Multiprogramació. Particions fixes. (III)

- Exemple: IBM OS/360, MFT (Multiprogramming with a Fixed number of Tasks). Un operador ho establia manualment.

Multiprogramació. Particions fixes. (III)

- Exemple: IBM OS/360, MFT (Multiprogramming with a Fixed number of Tasks). Un operador ho establia manualment.
- Inconvenients:

Multiprogramació. Particions fixes. (III)

- Exemple: IBM OS/360, MFT (Multiprogramming with a Fixed number of Tasks). Un operador ho establia manualment.
- Inconvenients:
  - Nombre de processos fixes.

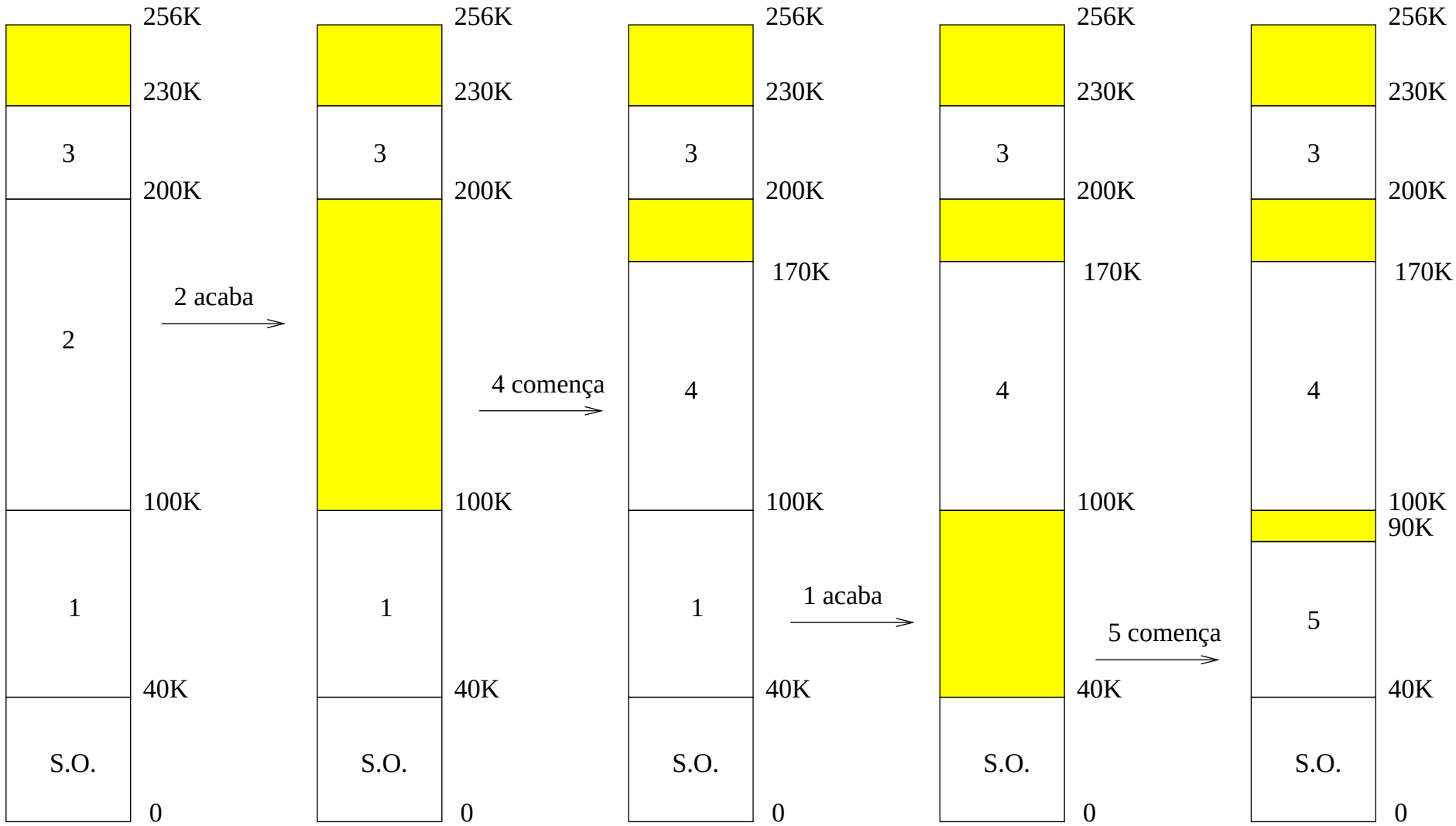
Multiprogramació. Particions fixes. (III)

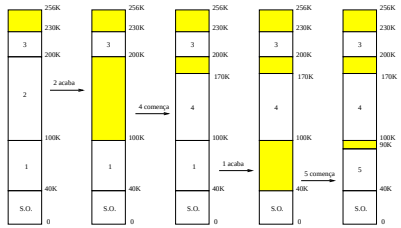
- Exemple: IBM OS/360, MFT (Multiprogramming with a Fixed number of Tasks). Un operador ho establia manualment.
- Inconvenients:
  - Nombre de processos fixes.
  - Fragmentació interna: Processos que no aprofiten la memòria disponible de la partició.

Multiprogramació. Particions fixes. (III)

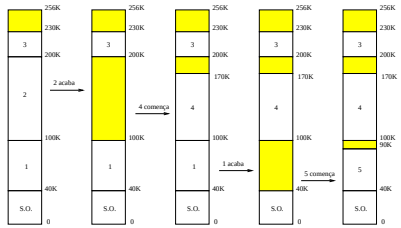
- Exemple: IBM OS/360, MFT (Multiprogramming with a Fixed number of Tasks). Un operador ho establia manualment.
- Inconvenients:
  - Nombre de processos fixes.
  - Fragmentació interna: Processos que no aprofiten la memòria disponible de la partició.
  - Fragmentació externa: Particions no usades ja que cap procés hi cap.

Multiprogramació. Particions dinàmiques (I).

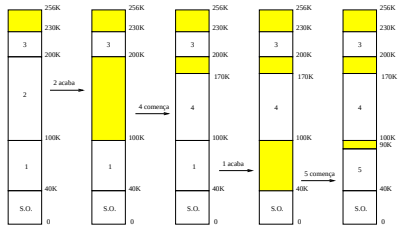




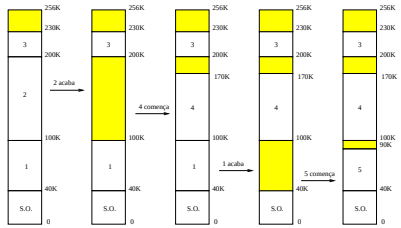
- En qualsevol moment tenim un conjunt de blocs lliures de diverses mides i dispersos en memòria.



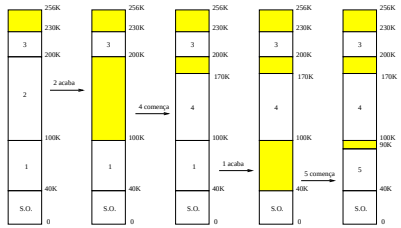
- En qualsevol moment tenim un conjunt de blocs lliures de diverses mides i dispersos en memòria.
- Quan un treball arriba i necessita memòria, hem de cercar quin bloc de memòria lliure i suficientment gran pel treball s'ha d'agafar. Si el bloc és llarg el dividim en 2.



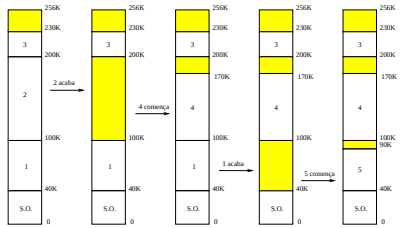
- En qualsevol moment tenim un conjunt de blocs lliures de diverses mides i dispersos en memòria.
- Quan un treball arriba i necessita memòria, hem de cercar quin bloc de memòria lliure i suficientment gran pel treball s'ha d'agafar. Si el bloc és llarg el dividim en 2.
- Quin bloc?



- En qualsevol moment tenim un conjunt de blocs lliures de diverses mides i dispersos en memòria.
- Quan un treball arriba i necessita memòria, hem de cercar quin bloc de memòria lliure i suficientment gran pel treball s'ha d'agafar. Si el bloc és llarg el dividim en 2.
- Quin bloc?
  - **First Fit.** Assignem el primer bloc que és suficientment gran i aturem la cerca. Algorisme ràpid.

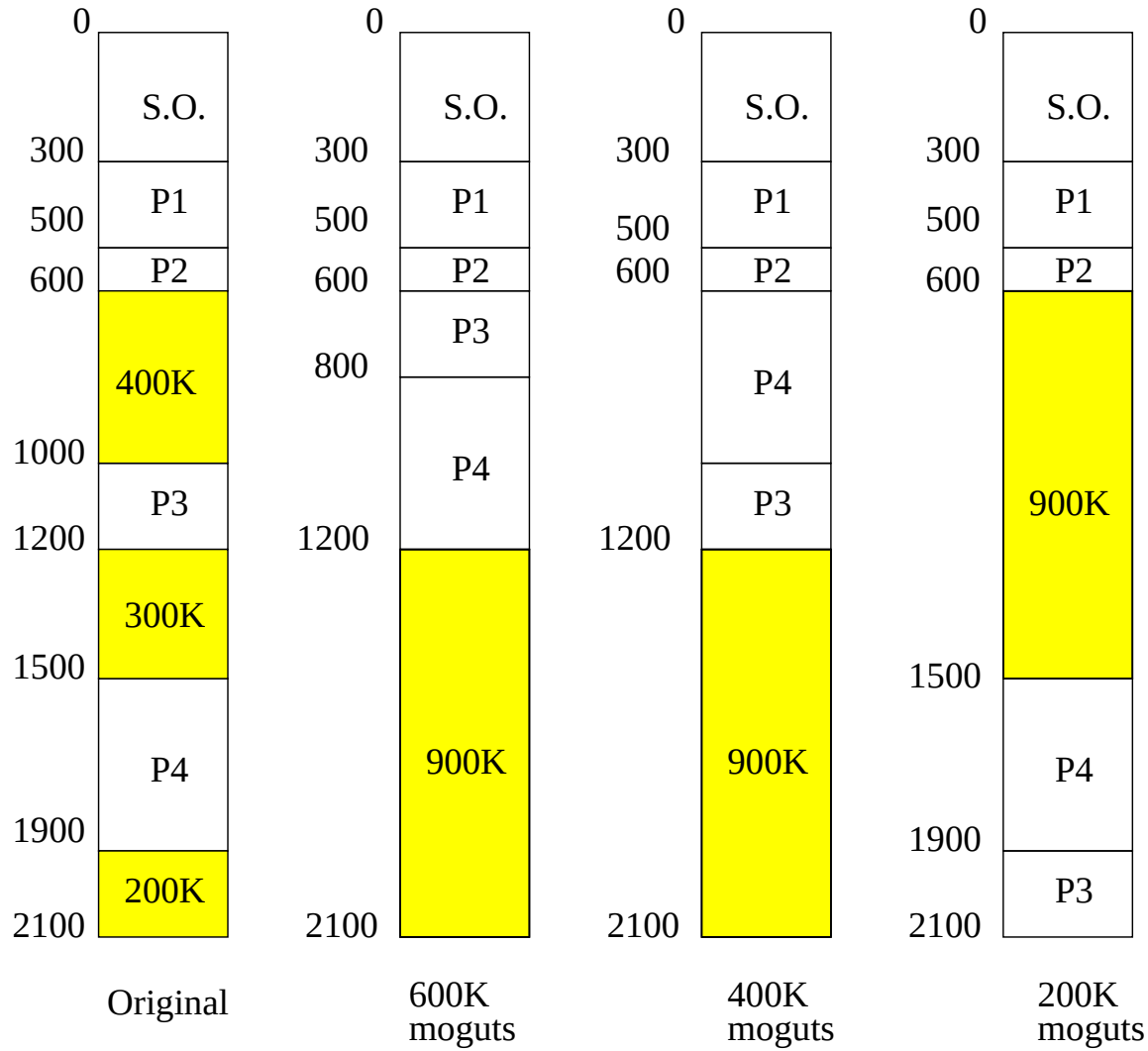


- En qualsevol moment tenim un conjunt de blocs lliures de diverses mides i dispersos en memòria.
- Quan un treball arriba i necessita memòria, hem de cercar quin bloc de memòria lliure i suficientment gran pel treball s'ha d'agafar. Si el bloc és llarg el dividim en 2.
- Quin bloc?
  - **First Fit.** Assignem el primer bloc que és suficientment gran i aturem la cerca. Algorisme ràpid.
  - **Best Fit.** El bloc més petit que és suficientment gran. Cal cercar per tota la llista de blocs.



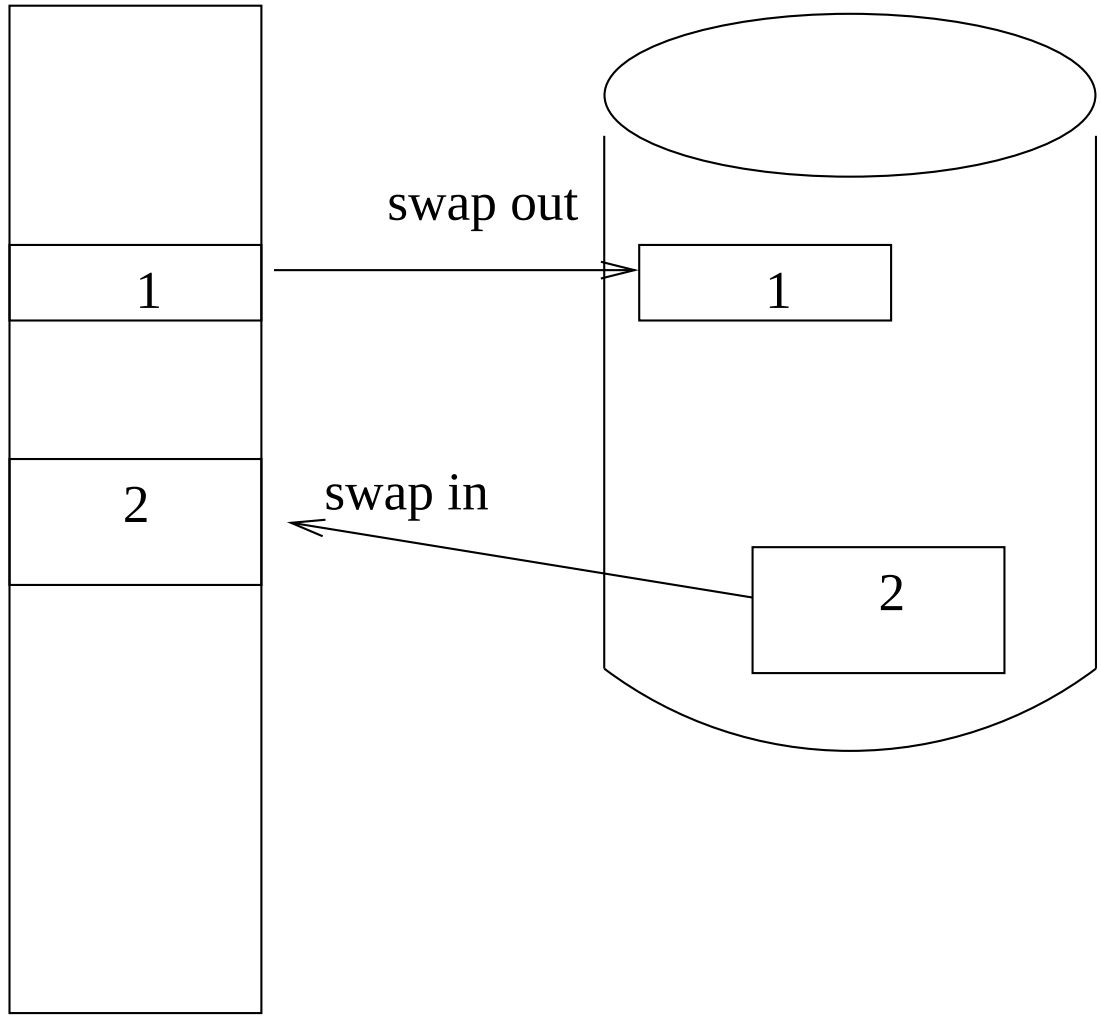
- En qualsevol moment tenim un conjunt de blocs lliures de diverses mides i dispersos en memòria.
- Quan un treball arriba i necessita memòria, hem de cercar quin bloc de memòria lliure i suficientment gran pel treball s'ha d'agafar. Si el bloc és llarg el dividim en 2.
- Quin bloc?
  - **First Fit.** Assignem el primer bloc que és suficientment gran i aturem la cerca. Algorisme ràpid.
  - **Best Fit.** El bloc més petit que és suficientment gran. Cal cercar per tota la llista de blocs.
  - **Worst Fit.** El bloc més gran que és suficientment gran. Cal cercar per tota la llista de blocs.

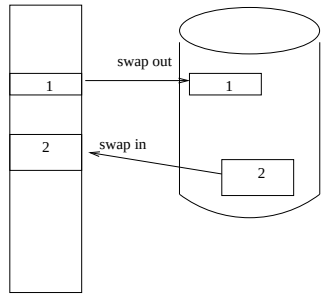
Compactació



Intercanvi (swapping).

Es guarden imatges de memòria de processos que no caben a la memòria principal al disc, i es baixen aquestes a la memòria principal quan calgui.

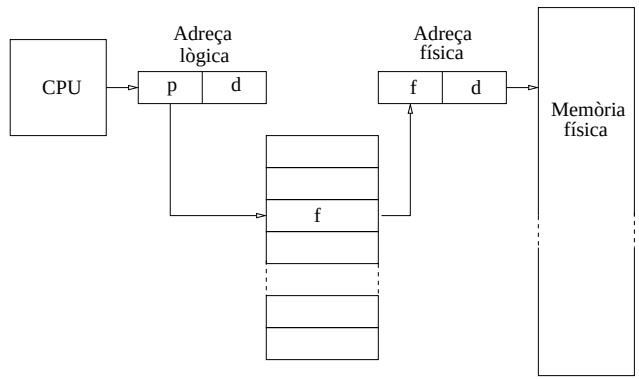




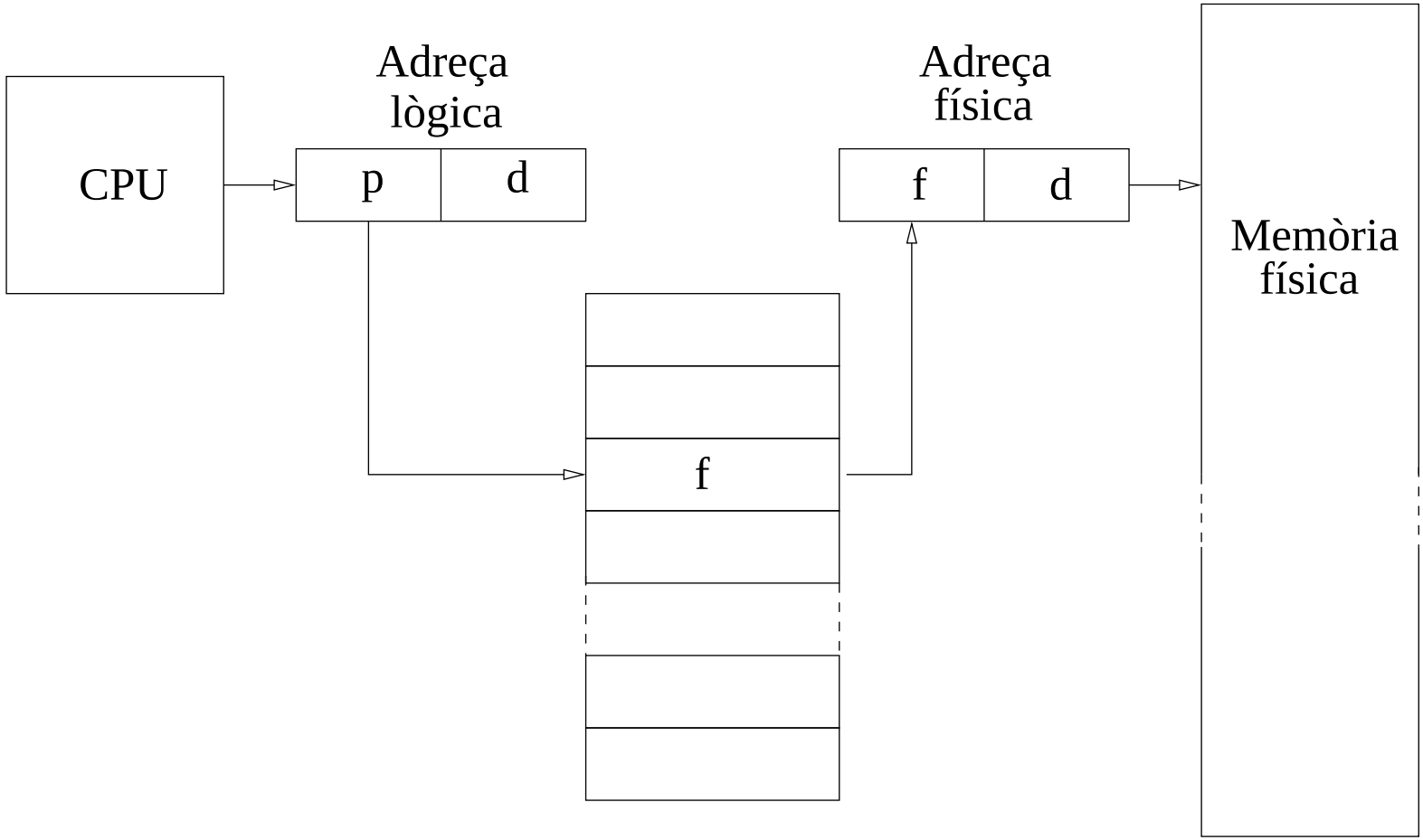
- Canvi de context costós en temps: proporcional a la velocitat del dispositiu (disc) i la mida de la transferència.
- Es pot superposar el temps d'intercanvi amb el d'execució d'un programa.
- Cal mantenir les proteccions de memòria.
- Si un procés espera una entrada/sortida, la memòria d'aquest pot ser intercanviada però cal que els buffers d'entrada/sortida es mantinguin a la memòria principal per atendre la petició. Una forma de fer-ho és que el sistema operatiu tingui els seus propis buffers.

Paginació (I).

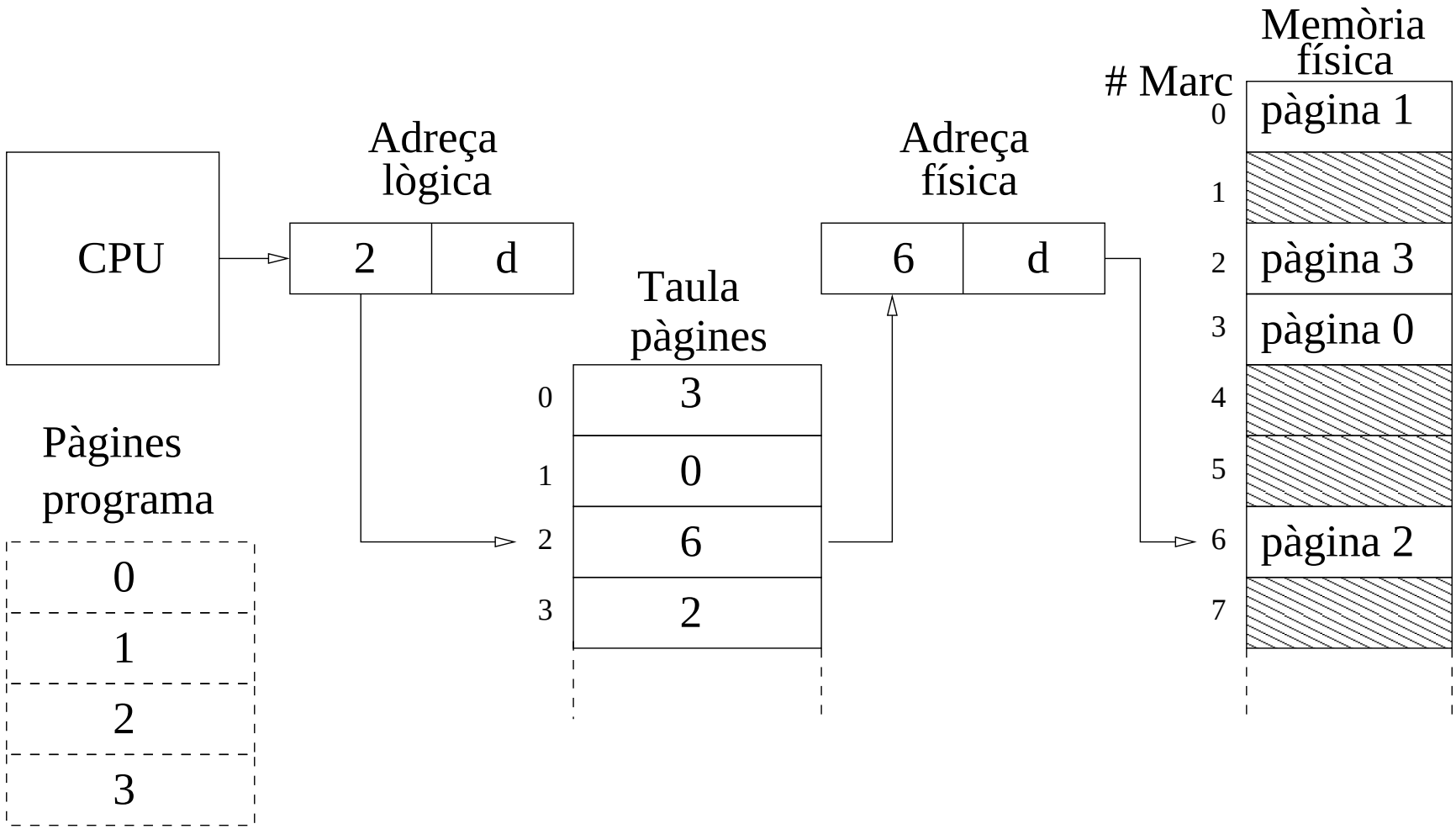
- Soluciona el problema de la fragmentació i fa que un programa no tingui que usar memòria contigua. D'aquesta forma s'aprofita gairebé tota la memòria.
- L'adreça generada pel processador es divideix en :
  - Nombre de **pàgina** (p): Indexa un element de la taula de pàgines que conté l'adreça base de la pàgina dins de la memòria física i que s'anomena **marc** (f).
  - Offset de pàgina (d). Combinat amb l'adreça base defineix la memòria física de la paraula a la que es vol accedir.
- Aporta una separació entre cóm veu l'usuari la memòria i la memòria física real. Mitjançant el hardware les adreces lògiques es tradueixen en adreces físiques.



Paginació (II).



Paginació (III). Exemple

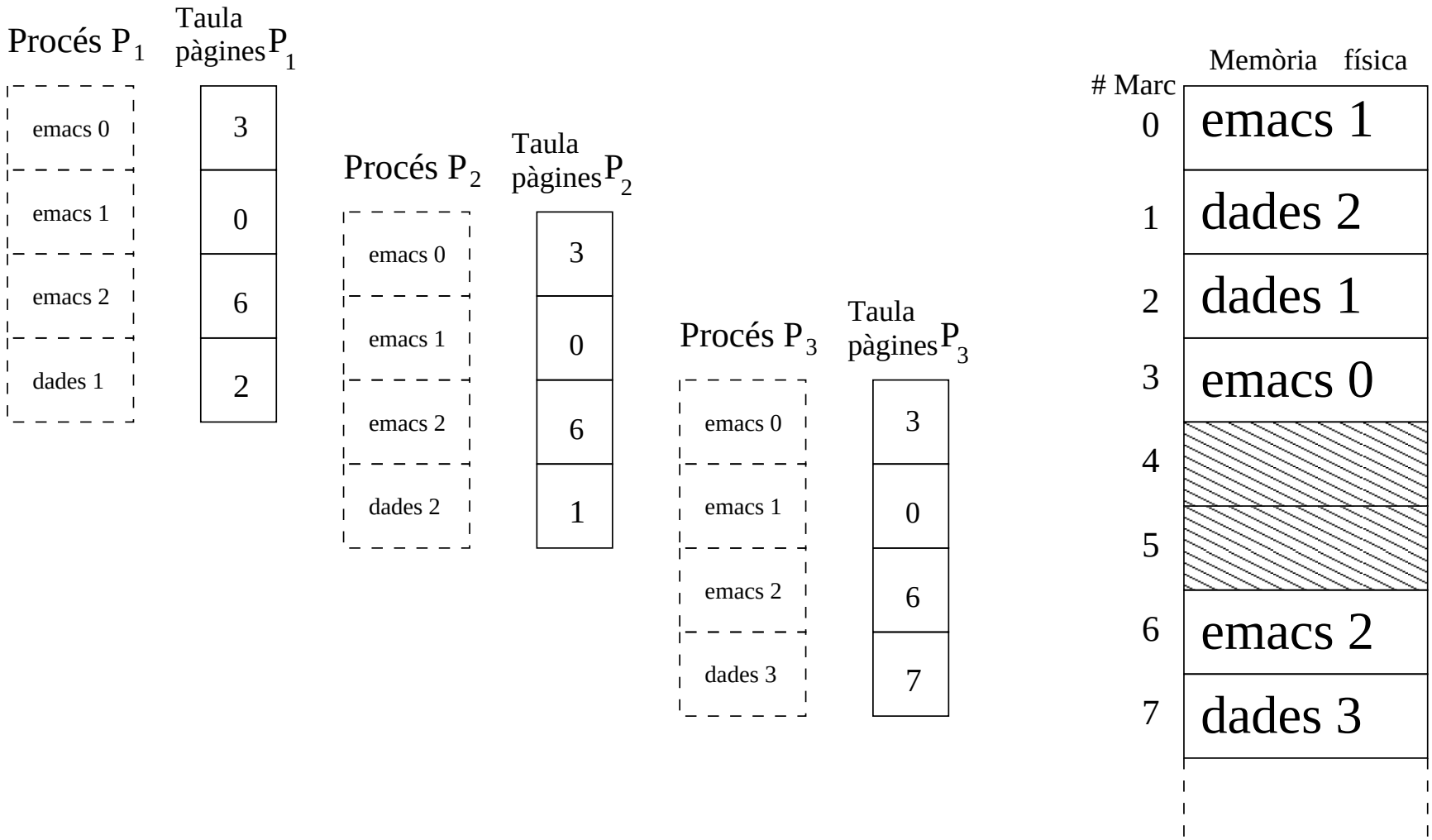


Paginació (IV).

- Soluciona el problema de la fragmentació i fa que un programa no tingui que usar memòria contigua. D'aquesta forma s'aprofita gairebé tota la memòria.
- La fragmentació externa s'elimina. La interna és manté ja que el darrer marc pot no estar ocupat del tot
- Cada procés tindrà la seva taula de pàgines.
- Tot això es suportat pel hardware mitjançant un xip que s'anomena MMU (Memory Management Unit)
- Es pot implementar la protecció de la memòria associant un bit de vàlid-invàlid a cada entrada de la taula. "Vàlid" indica que la pàgina associada és en l'espai d'adreces lògic, i per tant és una pàgina legal.

Paginació (V). Pàgines compartides

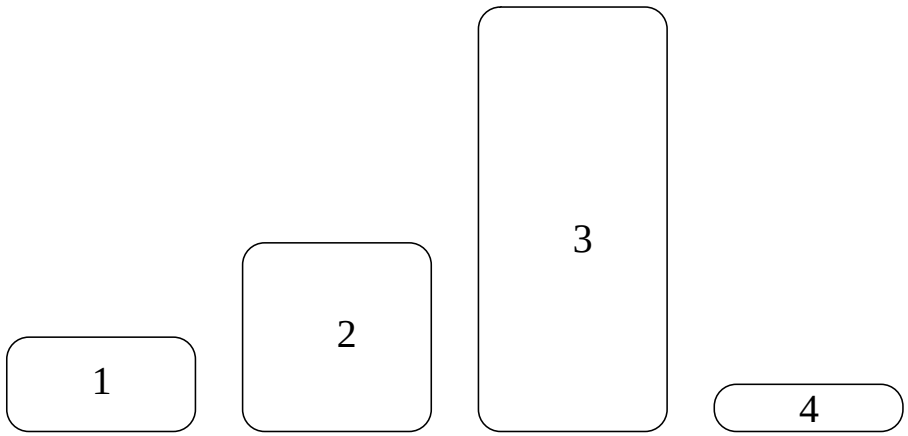
- Una còpia de codi (només lectura) compartida entre tots els processos.



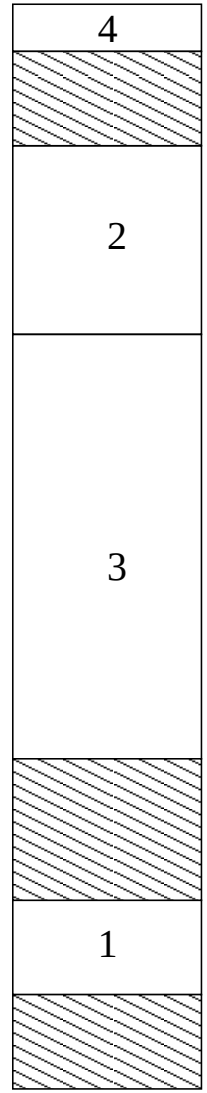
Segmentació.

- La segmentació és un esquema de gestió de memòria que suporta la visió de memòria de l'usuari
- Un programa és una col·lecció de segments. Un segment és una unitat lògica com
  - programa principal
  - acció
  - funció
  - variables locals/globals
  - blocs comuns
  - pila
  - taula de símbols, etc

Segmentació (II).



Espai usuari



Memòria física

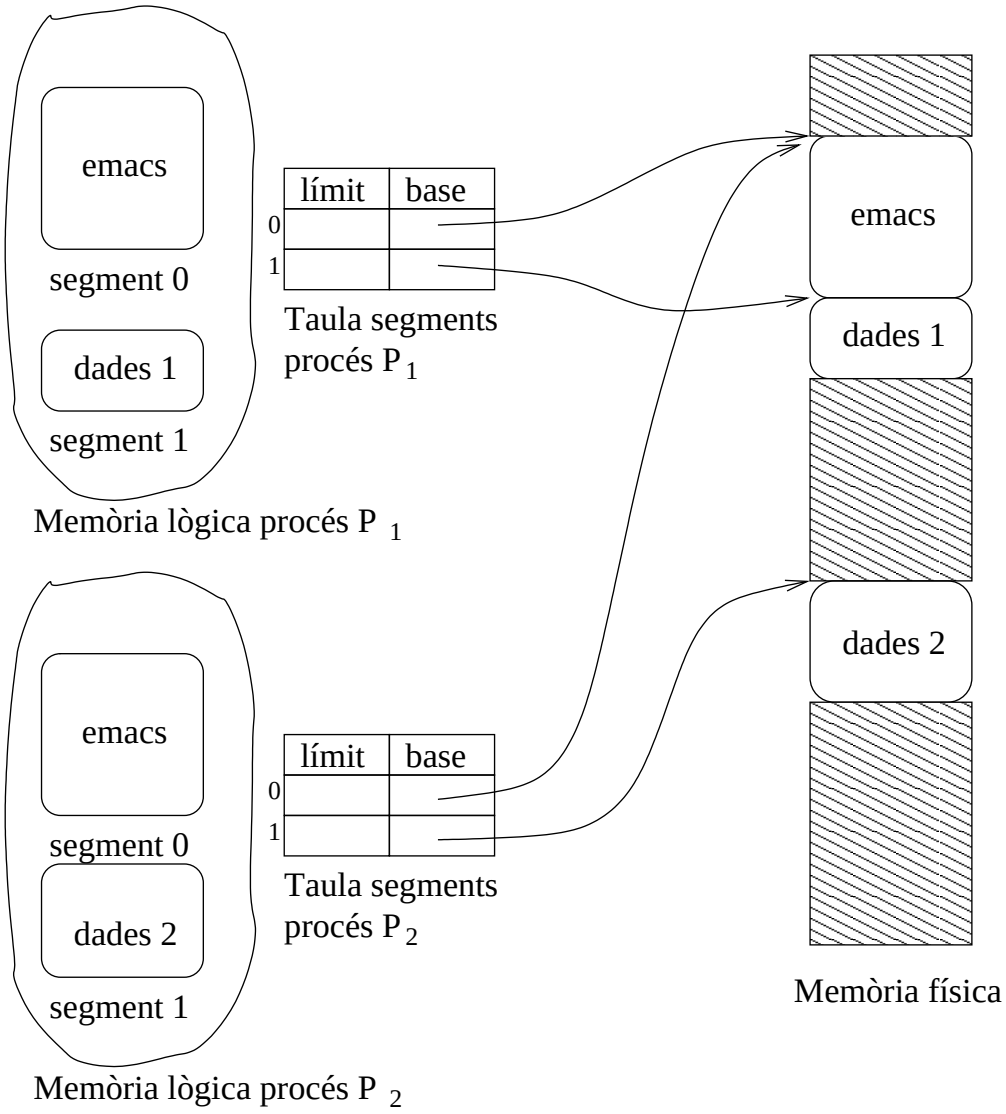
Segmentació (III).

- L'adreça lògica consistirà de

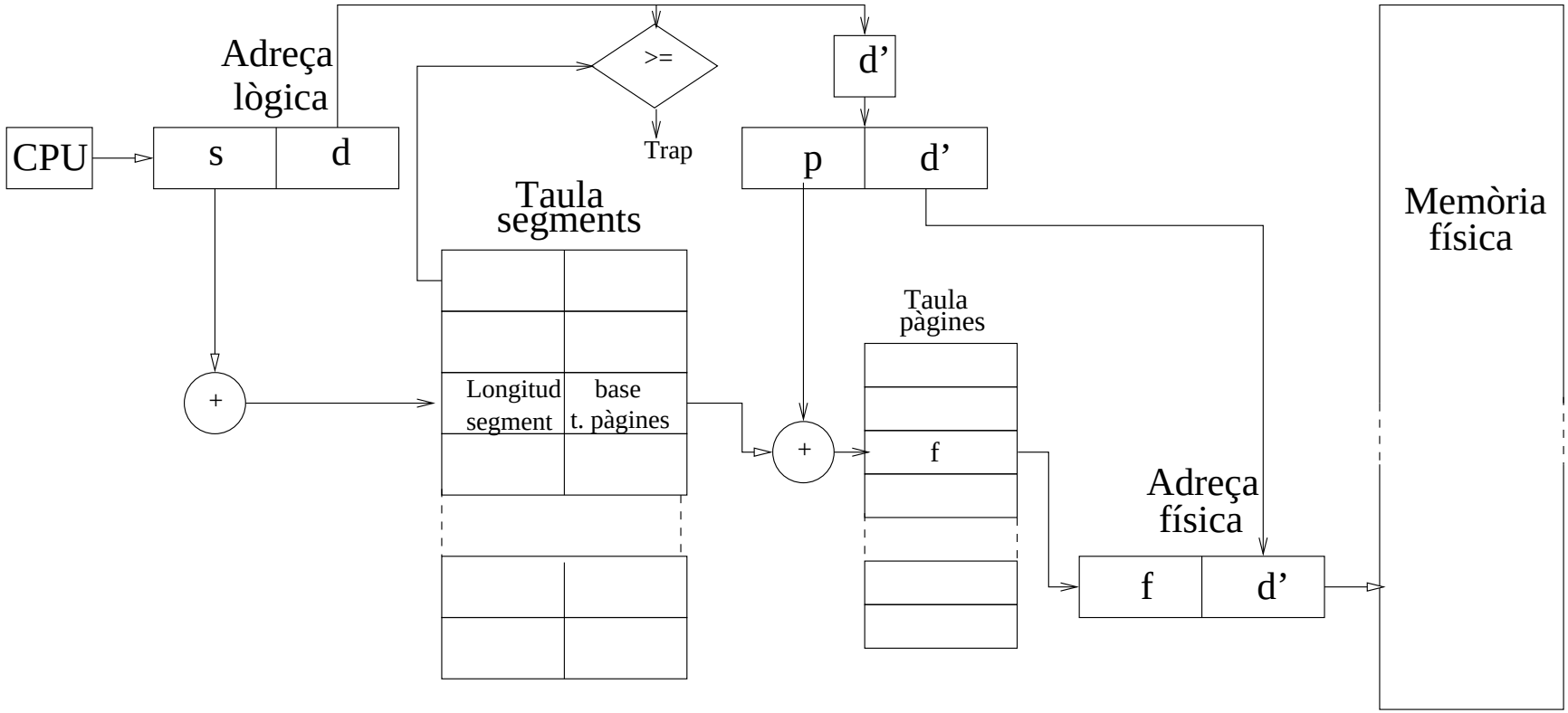
Nombre de segment	offset

- Una taula de segments fa correspondre l'adreça lògica en una de física. Cada entrada de la taula té:
  - base: Adreça inicial física on el segment resideix a la memòria.
  - límit: Longitud del segment.
- Un registre base de taula de segments $(STBR)$ apunta a la taula de segments.
- Un registre de longitud de taula de segments $(STLR)$ indica el nombre de segments del programa. Un segment serà legal si no sobrepassa aquest nombre.
- Des del punt de vista d'usuari, amb els segments podem: deslocalitzar-los dinàmicament via taula de segments, compartir-los amb altres processos, protegir-los (bit de validació a l'entrada de la taula, privilegis de lectura/escriptura,execució, i reservar memòria (first fit/best fit, fragmentació externa).

Segmentació.



Paginació amb segmentació. Exemple Multics



Consideracions sobre les estratègies de gestió de memòria

- Suport de hardware
- Rendiment
- Fragmentació
- Reubicació
- Intercanvi
- Compartició
- Protecció