

Gestió Memòria

Josep Vilaplana

ETSEIB/GIE

22 de maig de 2023

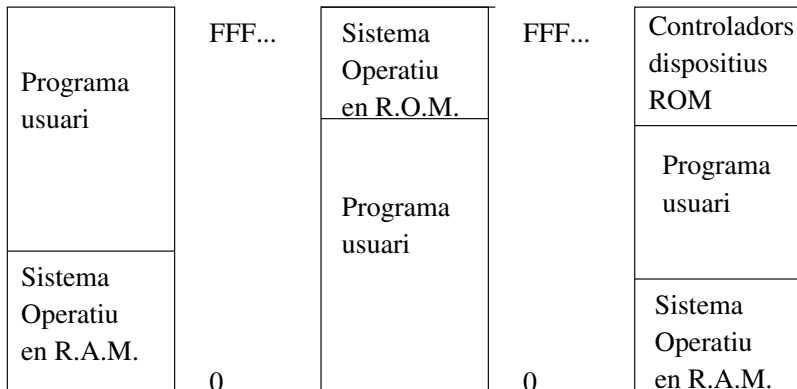
- Com organitzem la memòria de forma que els programes d'usuari i el sistema operatiu la puguin compartir?
- Com es protegeixen els diferents àmbits?
- Com vinculem les adreces amb el programa un cop sabem on han de residir?
- Què fem si tot no cap a la memòria principal?
- Què fem si un programa és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

- 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?

- 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?

- 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?

- 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?



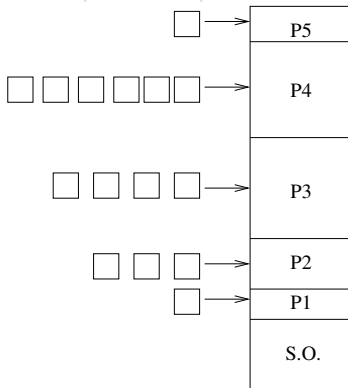
- La càrrega de programes d'usuari s'hauria de fer a partir de l'adreça del recinte.
- El compilador genera codi reubicable. El muntador i el servei que carrega el programa genera la ubicació definitiva.
- En la execució d'un programa l'adreça de recinte no pot variar. Si el sistema operatiu vol afegir 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.

- 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.

- 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.

- 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.

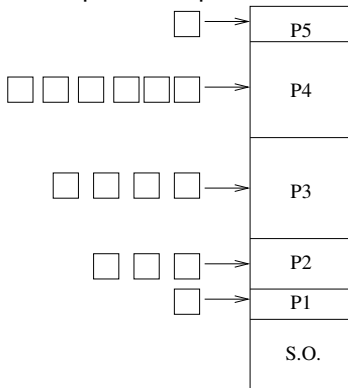
- Es divideix la memòria en n particions probablement desiguals.
- La planificació de processos haurà de tenir en compte si el procés cap en una partició donada.



Cues de processos

Memòria

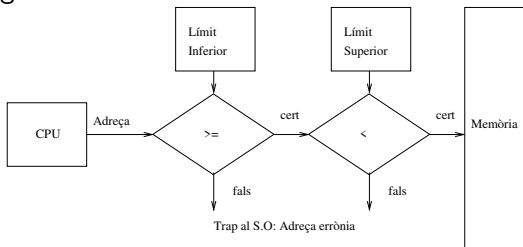
- Es divideix la memòria en n particions probablement desiguals.
- La planificació de processos haurà de tenir en compte si el procés cap en una partició donada.



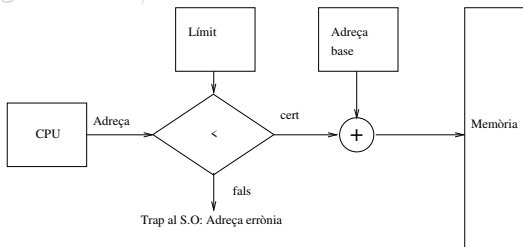
Cues de processos

Memòria

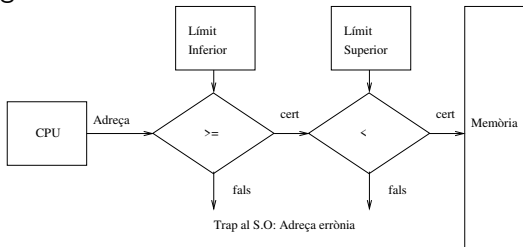
- Registres límits



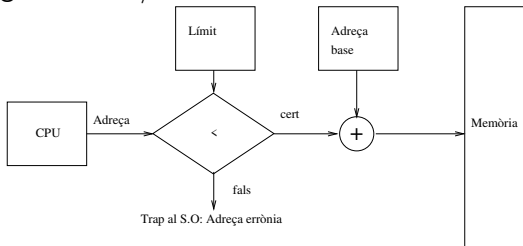
- Registres base/límit



- Registres límits



- Registres base/límit



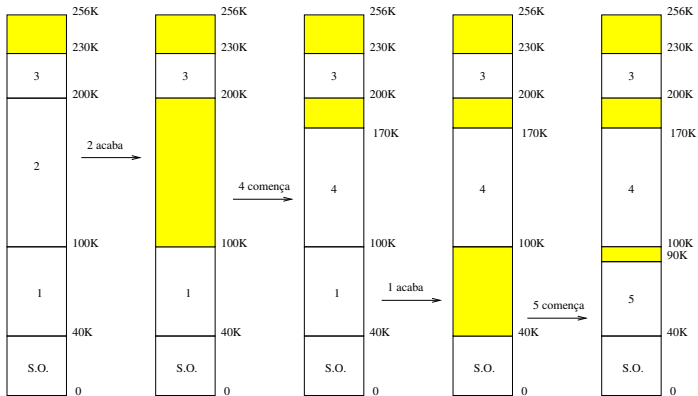
- Exemple: IBM OS/360, MFT (Multiprogramming with a Fixed number of Tasks). Un operador ho establiria manualment.
- Inconvenients:
 - Nombre de processos fixes.
 - Fragmentació interna: Processos que no aprofiten la memòria disponible de la partició.
 - Fragmentació externa: Particions no usades ja que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

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

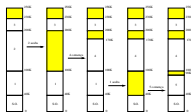
- Exemple: IBM OS/360, MFT (Multiprogramming with a Fixed number of Tasks). Un operador ho establia manualment.
- Inconvenients:
 - Nombre de processos fixes.
 - Fragmentació interna: Processos que no aprofiten la memòria disponible de la partició.
 - Fragmentació externa: Particions no usades ja que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

- 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.

- 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.

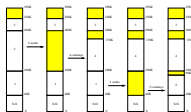


Multiprogramació. Particions dinàmiques (II).



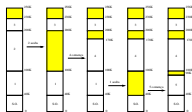
- En qualsevol moment tenim un conjunt de blocs lliures de diverses mides i dispersos en memòria.
- Quan un treball arriba i necessita memòria, hem de cercar quin bloc de memòria lliure i suficientment gran pel treball s'ha d'agafar. Si el bloc és llarg el dividim en 2.
- Quin bloc?
 - **First Fit.**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

Multiprogramació. Particions dinàmiques (II).



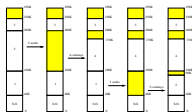
- En qualsevol moment tenim un conjunt de blocs lliures de diverses mides i dispersos en memòria.
- Quan un treball arriba i necessita memòria, hem de cercar quin bloc de memòria lliure i suficientment gran pel treball s'ha d'agafar. Si el bloc és llarg el dividim en 2.
- Quin bloc?
 - **First Fit.**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

Multiprogramació. Particions dinàmiques (II).



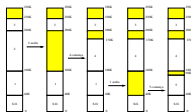
- En qualsevol moment tenim un conjunt de blocs lliures de diverses mides i dispersos en memòria.
- Quan un treball arriba i necessita memòria, hem de cercar quin bloc de memòria lliure i suficientment gran pel treball s'ha d'agafar. Si el bloc és llarg el dividim en 2.
- Quin bloc?
 - **First Fit.**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

Multiprogramació. Particions dinàmiques (II).



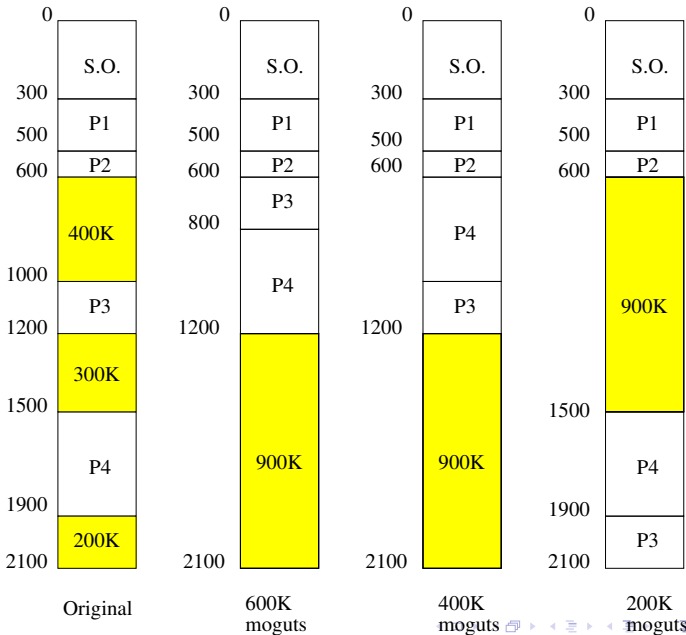
- En qualsevol moment tenim un conjunt de blocs lliures de diverses mides i dispersos en memòria.
- Quan un treball arriba i necessita memòria, hem de cercar quin bloc de memòria lliure i suficientment gran pel treball s'ha d'agafar. Si el bloc és llarg el dividim en 2.
- Quin bloc?
 - **First Fit.**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

Multiprogramació. Particions dinàmiques (II).



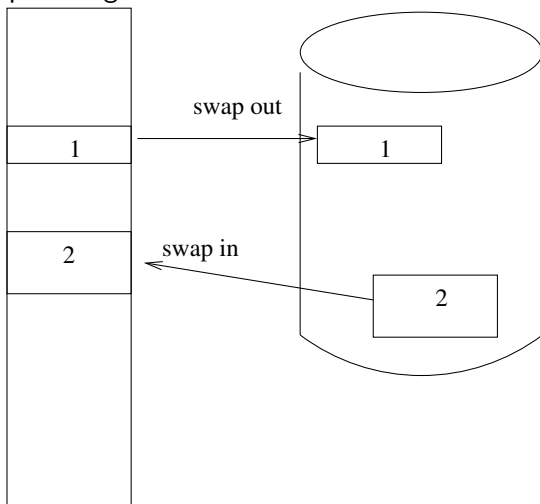
- En qualsevol moment tenim un conjunt de blocs lliures de diverses mides i dispersos en memòria.
- Quan un treball arriba i necessita memòria, hem de cercar quin bloc de memòria lliure i suficientment gran pel treball s'ha d'agafar. Si el bloc és llarg el dividim en 2.
- Quin bloc?
 - **First Fit.**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

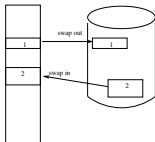


Intercanvi (swapping).

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.

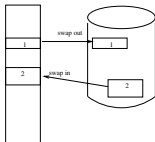


Intercanvi (swapping).



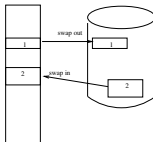
- Canvi de context costós en temps: proporcional a la velocitat del dispositiu (disc) i la mida de la transferència.
- Es pot superposar el temps d'intercanvi amb el d'execució d'un programa.
- Cal mantenir les proteccions de memòria.
- Si un procés espera una entrada/sortida, la memòria d'aquest pot 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.

Intercanvi (swapping).



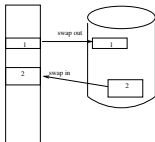
- Canvi de context costós en temps: proporcional a la velocitat del dispositiu (disc) i la mida de la transferència.
- Es pot superposar el temps d'intercanvi amb el d'execució d'un programa.
- Cal mantenir les proteccions de memòria.
- Si un procés espera una entrada/sortida, la memòria d'aquest pot 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.

Intercanvi (swapping).



- 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.

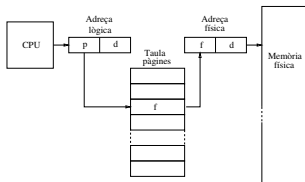
Intercanvi (swapping).



- 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.

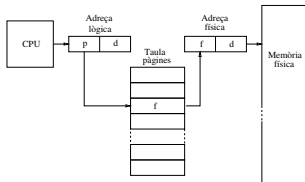
Paginació (I).

- Soluciona el problema de la fragmentació i fa que un programa no tingui que usar memòria contiguous. 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 com veu l'usuari la memòria i la memòria física real. Mitjançant el hardware les adreces lògiques es tradueixen en adreces físiques.



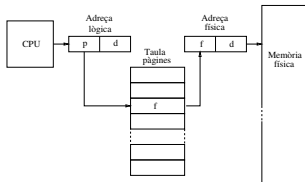
Paginació (I).

- Soluciona el problema de la fragmentació i fa que un programa no tingui que usar memòria contiguous. 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 com veu l'usuari la memòria i la memòria física real. Mitjançant el hardware les adreces lògiques es tradueixen en adreces físiques.



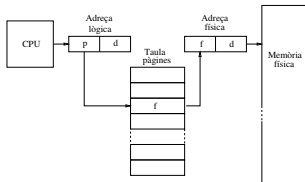
Paginació (I).

- Soluciona el problema de la fragmentació i fa que un programa no tingui que usar memòria contiguous. 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 com veu l'usuari la memòria i la memòria física real. Mitjançant el hardware les adreces lògiques es tradueixen en adreces físiques.



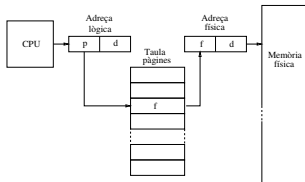
Paginació (I).

- Soluciona el problema de la fragmentació i fa que un programa no tingui que usar memòria contiguous. 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 com veu l'usuari la memòria i la memòria física real. Mitjançant el hardware les adreces lògiques es tradueixen en adreces físiques.

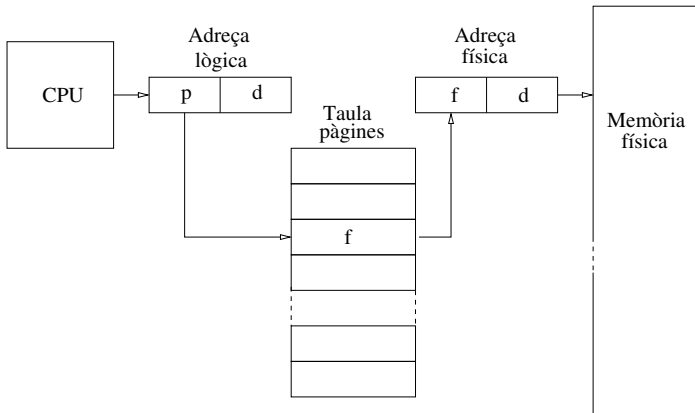


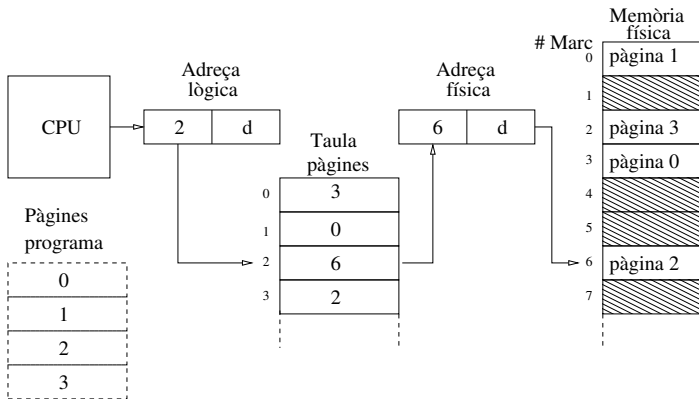
Paginació (I).

- Soluciona el problema de la fragmentació i fa que un programa no tingui que usar memòria contiguous. 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 com veu l'usuari la memòria i la memòria física real. Mitjançant el hardware les adreces lògiques es tradueixen en adreces físiques.



Paginació (II).





- 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.

- Soluciona el problema de la fragmentació i fa que un programa no tingui que usar memòria contiguous. 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 (IV).

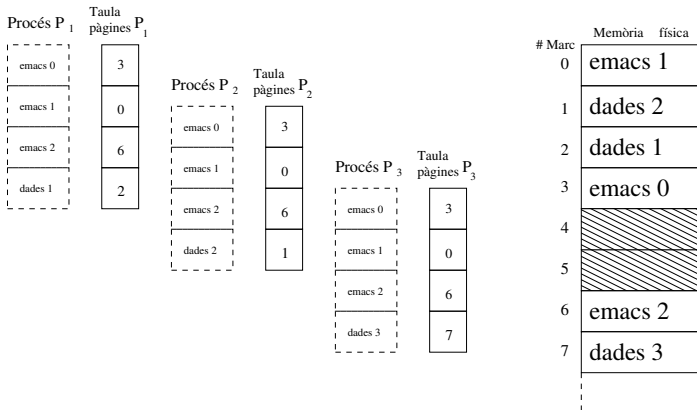
- Soluciona el problema de la fragmentació i fa que un programa no tingui que usar memòria contiguous. D'aquesta forma s'aprofita gairebé tota la memòria.
- La fragmentació externa s'elimina. La interna és manté ja que el darrer marc pot no estar ocupat del tot
- Cada procés tindrà la seva taula de pàgines.
- 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 (IV).

- Soluciona el problema de la fragmentació i fa que un programa no tingui que usar memòria contiguous. 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.

- Soluciona el problema de la fragmentació i fa que un programa no tingui que usar memòria contiguous. 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.

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



- 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

- 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

- 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

- 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

- 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

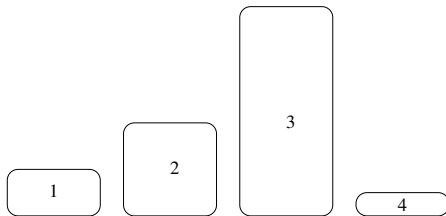
- La segmentació és un esquema de gestió de memòria que suporta la visió de memòria de l'usuari
- Un programa és una col·lecció de segments. Un segment és una unitat lògica com
 - programa principal
 - acció
 - funció
 - variables locals/globals
 - blocs comuns
 - pila
 - taula de símbols, etc

- 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

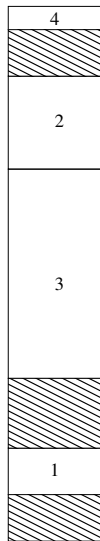
- La segmentació és un esquema de gestió de memòria que suporta la visió de memòria de l'usuari
- Un programa és una col·lecció de segments. Un segment és una unitat lògica com
 - programa principal
 - acció
 - funció
 - variables locals/globals
 - blocs comuns
 - pila
 - taula de símbols, etc

- 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 (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 (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 (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 (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 (III).

- L'adreça lògica consistirà de

Nombre de segment	offset
-------------------	--------

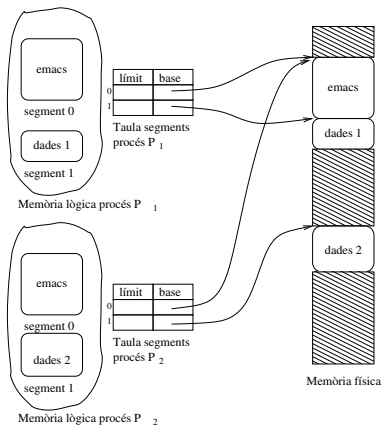
- Una taula de segments fa correspondre l'adreça lògica en una de física. Cada entrada de la taula té:
 - base: Adreça inicial física on el segment resideix a la memòria.
 - límit: Longitud del segment.
- Un registre base de taula de segments (*STBR*) apunta a la taula de segments.
- Un registre de longitud 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).

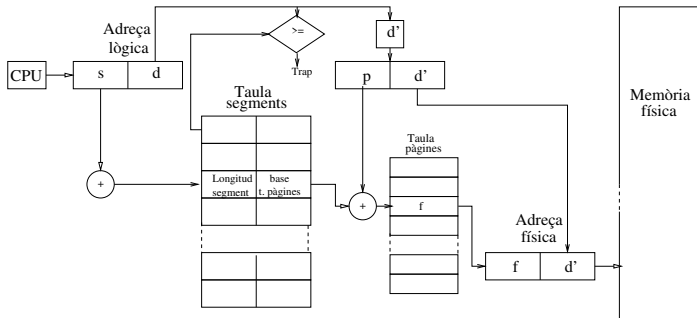
Segmentació (III).

- 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).





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

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

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

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

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

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

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

- Memòria Virtual:

- Separa la memòria lògica de l'usuari de la memòria física.
- Permet executar un programa tenint només una part del programa a memòria, ja que no cal tenir tot el programa a memòria per executar una part.
- Permet que l'espai d'**adreces lògiques** pugui ser més gran que l'espai d'**adreces físiques**.
- Permet un **intercanvi (swapping)** de pàgines entre memòria primària i secundària.

- Conceptes clau per fer-ho:

- Una tècnica per portar les pàgines de la memòria secundària a la primària. Presentarem la **paginació per demanda**.
- Conjunt de treball o **working set**: col·lecció mínima de pàgines que han d'estar present a la memòria principal per assegurar l'execució eficient del programa.

- Memòria Virtual:

- Separa la memòria lògica de l'usuari de la memòria física.
- Permet executar un programa tenint només una part del programa a memòria, ja que no cal tenir tot el programa a memòria per executar una part.
- Permet que l'espai d'**adreces lògiques** pugui ser més gran que l'espai d'**adreces físiques**.
- Permet un **intercanvi (swapping)** de pàgines entre memòria primària i secundària.

- Conceptes clau per fer-ho:

- Una tècnica per portar les pàgines de la memòria secundària a la primària. Presentarem la **paginació per demanda**.
- Conjunt de treball o **working set**: col·lecció mínima de pàgines que han d'estar present a la memòria principal per assegurar l'execució eficient del programa.

- Memòria Virtual:
 - Separa la memòria lògica de l'usuari de la memòria física.
 - Permet executar un programa tenint només una part del programa a memòria, ja que no cal tenir tot el programa a memòria per executar una part.
 - Permet que l'espai d'**adreces lògiques** pugui ser més gran que l'espai d'**adreces físiques**.
 - Permet un **intercanvi (swapping)** de pàgines entre memòria primària i secundària.
- Conceptes clau per fer-ho:
 - Una tècnica per portar les pàgines de la memòria secundària a la primària. Presentarem la **paginació per demanda**.
 - Conjunt de treball o **working set**: col·lecció mínima de pàgines que han d'estar present a la memòria principal per assegurar l'execució eficient del programa.

- Memòria Virtual:
 - Separa la memòria lògica de l'usuari de la memòria física.
 - Permet executar un programa tenint només una part del programa a memòria, ja que no cal tenir tot el programa a memòria per executar una part.
 - Permet que l'espai d'**adreces lògiques** pugui ser més gran que l'espai d'**adreces físiques**.
 - Permet un **intercanvi (swapping)** de pàgines entre memòria primària i secundària.
- Conceptes clau per fer-ho:
 - Una tècnica per portar les pàgines de la memòria secundària a la primària. Presentarem la **paginació per demanda**.
 - Conjunt de treball o **working set**: col·lecció mínima de pàgines que han d'estar present a la memòria principal per assegurar l'execució eficient del programa.

- Memòria Virtual:
 - Separa la memòria lògica de l'usuari de la memòria física.
 - Permet executar un programa tenint només una part del programa a memòria, ja que no cal tenir tot el programa a memòria per executar una part.
 - Permet que l'espai d'**adreces lògiques** pugui ser més gran que l'espai d'**adreces físiques**.
 - Permet un **intercanvi (swapping)** de pàgines entre memòria primària i secundària.
- Conceptes clau per fer-ho:
 - Una tècnica per portar les pàgines de la memòria secundària a la primària. Presentarem la **paginació per demanda**.
 - Conjunt de treball o **working set**: col·lecció mínima de pàgines que han d'estar present a la memòria principal per assegurar l'execució eficient del programa.

- Memòria Virtual:
 - Separa la memòria lògica de l'usuari de la memòria física.
 - Permet executar un programa tenint només una part del programa a memòria, ja que no cal tenir tot el programa a memòria per executar una part.
 - Permet que l'espai d'**adreces lògiques** pugui ser més gran que l'espai d'**adreces físiques**.
 - Permet un **intercanvi (swapping)** de pàgines entre memòria primària i secundària.
- Conceptes clau per fer-ho:
 - Una tècnica per portar les pàgines de la memòria secundària a la primària. Presentarem la **paginació per demanda**.
 - Conjunt de treball o **working set**: col·lecció mínima de pàgines que han d'estar present a la memòria principal per assegurar l'execució eficient del programa.

- Memòria Virtual:
 - Separa la memòria lògica de l'usuari de la memòria física.
 - Permet executar un programa tenint només una part del programa a memòria, ja que no cal tenir tot el programa a memòria per executar una part.
 - Permet que l'espai d'**adreces lògiques** pugui ser més gran que l'espai d'**adreces físiques**.
 - Permet un **intercanvi (swapping)** de pàgines entre memòria primària i secundària.
- Conceptes clau per fer-ho:
 - Una tècnica per portar les pàgines de la memòria secundària a la primària. Presentarem la **paginació per demanda**.
 - Conjunt de treball o **working set**: col·lecció mínima de pàgines que han d'estar present a la memòria principal per assegurar l'execució eficient del programa.

- Memòria Virtual:
 - Separa la memòria lògica de l'usuari de la memòria física.
 - Permet executar un programa tenint només una part del programa a memòria, ja que no cal tenir tot el programa a memòria per executar una part.
 - Permet que l'espai d'**adreces lògiques** pugui ser més gran que l'espai d'**adreces físiques**.
 - Permet un **intercanvi (swapping)** de pàgines entre memòria primària i secundària.
- Conceptes clau per fer-ho:
 - Una tècnica per portar les pàgines de la memòria secundària a la primària. Presentarem la **paginació per demanda**.
 - Conjunt de treball o **working set**: col·lecció mínima de pàgines que han d'estar present a la memòria principal per assegurar l'execució eficient del programa.

- Porta una pàgina a la memòria quan calgui:
 - Es necessiten menys entrades/sortides entre memòries
 - Menys memòria necessària
 - Resposta més ràpida (El programa comença sense carregar totes les pàgines)
 - Caben més processos
- Quan cal una pàgina $\Rightarrow$ la referenciem
 - referència invàlida $\Rightarrow$ atura el procés
 - fallida de pàgina: no està a memòria $\Rightarrow$ porta-la a memòria

- Porta una pàgina a la memòria quan calgui:
 - Es necessiten menys entrades/sortides entre memòries
 - Menys memòria necessària
 - Resposta més ràpida (El programa comença sense carregar totes les pàgines)
 - Caben més processos
- Quan cal una pàgina $\Rightarrow$ la referenciem
 - referència invàlida $\Rightarrow$ atura el procés
 - fallida de pàgina: no està a memòria $\Rightarrow$ porta-la a memòria

- Porta una pàgina a la memòria quan calgui:
 - Es necessiten menys entrades/sortides entre memòries
 - Menys memòria necessària
 - Resposta més ràpida (El programa comença sense carregar totes les pàgines)
 - Caben més processos
- Quan cal una pàgina $\Rightarrow$ la referenciem
 - referència invàlida $\Rightarrow$ atura el procés
 - fallida de pàgina: no està a memòria $\Rightarrow$ porta-la a memòria

- Porta una pàgina a la memòria quan calgui:
 - Es necessiten menys entrades/sortides entre memòries
 - Menys memòria necessària
 - Resposta més ràpida (El programa comença sense carregar totes les pàgines)
 - Caben més processos
- Quan cal una pàgina $\Rightarrow$ la referenciem
 - referència invàlida $\Rightarrow$ atura el procés
 - fallida de pàgina: no està a memòria $\Rightarrow$ porta-la a memòria

- Porta una pàgina a la memòria quan calgui:
 - Es necessiten menys entrades/sortides entre memòries
 - Menys memòria necessària
 - Resposta més ràpida (El programa comença sense carregar totes les pàgines)
 - Caben més processos
- Quan cal una pàgina $\Rightarrow$ la referenciem
 - referència invàlida $\Rightarrow$ atura el procés
 - fallida de pàgina: no està a memòria $\Rightarrow$ porta-la a memòria

- Porta una pàgina a la memòria quan calgui:
 - Es necessiten menys entrades/sortides entre memòries
 - Menys memòria necessària
 - Resposta més ràpida (El programa comença sense carregar totes les pàgines)
 - Caben més processos
- Quan cal una pàgina $\Rightarrow$ la referenciem
 - referència invàlida $\Rightarrow$ atura el procés
 - fallida de pàgina: no està a memòria $\Rightarrow$ porta-la a memòria

- Porta una pàgina a la memòria quan calgui:
 - Es necessiten menys entrades/sortides entre memòries
 - Menys memòria necessària
 - Resposta més ràpida (El programa comença sense carregar totes les pàgines)
 - Caben més processos
- Quan cal una pàgina $\Rightarrow$ la referenciem
 - referència invàlida $\Rightarrow$ atura el procés
 - fallida de pàgina: no està a memòria $\Rightarrow$ porta-la a memòria

- Porta una pàgina a la memòria quan calgui:
 - Es necessiten menys entrades/sortides entre memòries
 - Menys memòria necessària
 - Resposta més ràpida (El programa comença sense carregar totes les pàgines)
 - Caben més processos
- Quan cal una pàgina $\Rightarrow$ la referenciem
 - referència invàlida $\Rightarrow$ atura el procés
 - fallida de pàgina: no està a memòria $\Rightarrow$ porta-la a memòria

Memòria virtual. Paginació per demanda

0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H

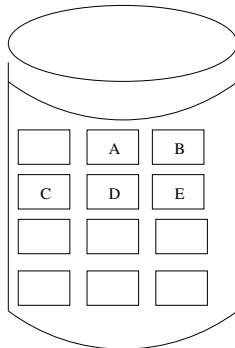
Memòria
Lògica

0	4	v
1		i
2	6	v
3		i
4		i
5	9	v
6		i
7		i

Taula
Pàgines

0	
1	
2	
3	
4	A
5	
6	C
7	
8	
9	F
10	
11	
12	
13	
14	
15	

Memòria
Física



Memòria
Secundària

- El working set (conjunt de treball) és la col·lecció mínima d'informació que ha d'estar present a la memòria principal per assegurar l'execució eficient del programa.
- Des del punt de vista del sistema, es tradueix pel conjunt de pàgines molt recentment referenciades.
- Com que no cal tenir tot a memòria per executar un procés, l'ús mínim de memòria per treballar permetrà que altres processos puguin estar a memòria. Augmenta el grau de multiprogramació.
- Un dels paràmetres importants és la mida del working set que hauria de minimitzar la fallida de pàgines. Cal observar i aprofitar la proximitat de les referències del procés. Per exemple,

5 4 3 1 6 6 6 6 1 4 1 2 7 5 3 0 3 2 2 2 2 7 7 7 7 2 2 7

Working set: 3, 5, 1, 4, 6

Working set: 2, 7

- El working set (conjunt de treball) és la col·lecció mínima d'informació que ha d'estar present a la memòria principal per assegurar l'execució eficient del programa.
- Des del punt de vista del sistema, es tradueix pel conjunt de pàgines molt recentment referenciades.
- Com que no cal tenir tot a memòria per executar un procés, l'ús mínim de memòria per treballar permetrà que altres processos puguin estar a memòria. Augmenta el grau de multiprogramació.
- Un dels paràmetres importants és la mida del working set que hauria de minimitzar la fallida de pàgines. Cal observar i aprofitar la proximitat de les referències del procés. Per exemple,

5 4 3 1 6 6 6 6 1 4 1 2 7 5 3 0 3 2 2 2 2 7 7 7 7 2 2 7

Working set: 3, 5, 1, 4, 6

Working set: 2, 7

Quines pàgines del working set substituïm quan no caben més?
Per exemple, LRU (Least Recently Used)

5	7	5	3	1	5	2	1	7	5	1	5
5	5	5	5	5	5	5	5	7	7	7	7
	7	7	7	1	1	1	1	1	1	1	1
			3	3	3	2	2	2	5	5	5
F	F		F	F		F		F	F		

- **Thrashing (Pallissa):** El procés perd més temps paginant que executant-se
- Per exemple:
 - Es tenen processos amb la mida del conjunt de treball massa petita. Això fa augmentar la fallida de pàgines.
 - La fallida de pàgines augmenta les entrades/sortides de memòria secundària. La CPU s'usa poc doncs ha d'esperar que les entrades/sortides siguin satisfetes.
 - El planificador està configurat per admetre més processos si la CPU té poca feina.
 - Entren més processos agreujant la situació si les mides del seu conjunt de treball no són adients.

- **Thrashing (Pallissa):** El procés perd més temps paginant que executant-se
- Per exemple:
 - Es tenen processos amb la mida del conjunt de treball massa petita. Això fa augmentar la fallida de pàgines.
 - La fallida de pàgines augmenta les entrades/sortides de memòria secundària. La CPU s'usa poc doncs ha d'esperar que les entrades/sortides siguin satisfetes.
 - El planificador està configurat per admetre més processos si la CPU té poca feina.
 - Entren més processos agreujant la situació si les mides del seu conjunt de treball no són adients.

- **Thrashing (Pallissa):** El procés perd més temps paginant que executant-se
- Per exemple:
 - Es tenen processos amb la mida del conjunt de treball massa petita. Això fa augmentar la fallida de pàgines.
 - La fallida de pàgines augmenta les entrades/sortides de memòria secundària. La CPU s'usa poc doncs ha d'esperar que les entrades/sortides siguin satisfetes.
 - El planificador està configurat per admetre més processos si la CPU té poca feina.
 - Entren més processos agreujant la situació si les mides del seu conjunt de treball no són adients.

- **Thrashing (Pallissa):** El procés perd més temps paginant que executant-se
- Per exemple:
 - Es tenen processos amb la mida del conjunt de treball massa petita. Això fa augmentar la fallida de pàgines.
 - La fallida de pàgines augmenta les entrades/sortides de memòria secundària. La CPU s'usa poc doncs ha d'esperar que les entrades/sortides siguin satisfetes.
 - El planificador està configurat per admetre més processos si la CPU té poca feina.
 - Entren més processos agreujant la situació si les mides del seu conjunt de treball no són adients.

- **Thrashing (Pallissa):** El procés perd més temps paginant que executant-se
- Per exemple:
 - Es tenen processos amb la mida del conjunt de treball massa petita. Això fa augmentar la fallida de pàgines.
 - La fallida de pàgines augmenta les entrades/sortides de memòria secundària. La CPU s'usa poc doncs ha d'esperar que les entrades/sortides siguin satisfetes.
 - El planificador està configurat per admetre més processos si la CPU té poca feina.
 - Entren més processos agreujant la situació si les mides del seu conjunt de treball no són adients.

- **Thrashing (Pallissa):** El procés perd més temps paginant que executant-se
- Per exemple:
 - Es tenen processos amb la mida del conjunt de treball massa petita. Això fa augmentar la fallida de pàgines.
 - La fallida de pàgines augmenta les entrades/sortides de memòria secundària. La CPU s'usa poc doncs ha d'esperar que les entrades/sortides siguin satisfetes.
 - El planificador està configurat per admetre més processos si la CPU té poca feina.
 - Entren més processos agreujant la situació si les mides del seu conjunt de treball no són adients.