

Tema 7. Memòria Virtual

Estructura de Computadors (EC)

Rubèn Tous

rtous@ac.upc.edu
Computer Architecture Department
Universitat Politècnica de Catalunya



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Índex

- 1 7.1 Introducció
- 2 7.2 Funcionament de la memòria virtual
- 3 7.3 Fallada de pàgina
- 4 7.4 Traducció ràpida amb TLB
- 5 7.5 Protecció i Compartició

7.1.1 Motivació

- El procés d'enllaçat assigna adreces de memòria absolutes.
- Però es poden executar múltiples programes simultàniament. Com és possible saber sobre quines adreces de memòria treballarà un programa abans d'executar-lo?
- I si el programa supera la grandària de la memòria física?

7.1.1 Motivació

- Resposta: *memòria virtual*.
- Per un costat, permet que la memòria sigui compartida de manera eficient i segura per múltiples programes.
- Per un altre costat, permet a un o més programes excedir la capacitat de la MP, gracies a l'ús de l'emmagatzematge secundari.

7.1.2 Espai d'adreçament físic i espai d'adreçament lògic

- L'MV consisteix essencialment en fer servir dos espais d'adreçament diferents:
 - L'*espai d'adreçament lògic* o virtual: les adreces que hi haurà als programes (quan els compilem i també quan els executem).
 - L'*espai d'adreçament físic*: adreces de la memòria física (les adreces reals").

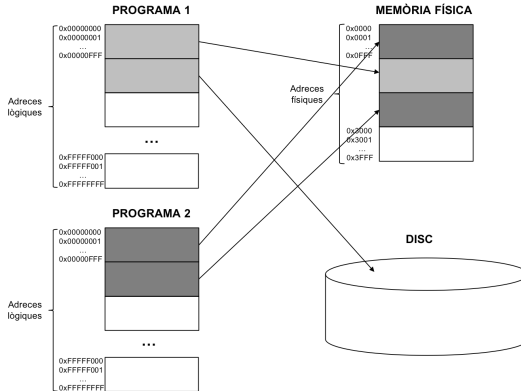
7.1.2 Espai d'adreçament físic i espai d'adreçament lògic

- L'*espai d'adreçament lògic* o virtual (MV):
 - Exclusiu de cada programa.
 - Grandària: la màxima permesa pel número de bits d'adreça.
 - Aïlla de la complexitat real: impressió de que només hi ha un programa i memòria il·limitada.
 - Les adreces del compilador i de la CPU: adreces lògiques.

7.1.2 Espai d'adreçament físic i espai d'adreçament lògic

- *L'espai d'adreçament físic (MF):*
 - Adreces de la memòria física (les adreces reals").
 - Durant l'execució d'un programa, caldrà carregar-lo a l'MF.
 - L'MV defineix un mecanisme que permet recordarà quines adreces físiques s'han carregat les adreces lògiques d'un programa.
 - Quan la CPU sol·licita una dada, l'adreça lògica es tradueixen l'adreça física on realment és la dada.

7.1.2 Espai d'adreçament físic i espai d'adreçament lògic

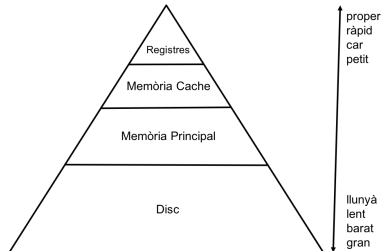


7.1.3 El disc com un nivell més en la jerarquia de memòria

- Una de les motivacions originals de l'MV: límitació de la memòria física.
- Abans es solucionava mitjançant *overlaying*.
- L'MV funciona d'una manera semblant, posarem al disc les parts d'un programa que no s'estan fent servir.
- Ho gestiona el sistema operatiu, amb l'ajuda del hardware.

7.1.3 El disc com un nivell més en la jerarquia de memòria

- El disc passa a ser una capa més de la jerarquia de memòria.
- La relació entre el disc i l'MF serà la mateixa que hi ha entre l'MP i l'MC.



Índex

- 1 7.1 Introducció
- 2 7.2 Funcionament de la memòria virtual
- 3 7.3 Fallada de pàgina
- 4 7.4 Traducció ràpida amb TLB
- 5 7.5 Protecció i Compartició

7.2.1 Unitat bàsica de gestió de memòria: la pàgina

- Terminologia diferent que a MC però mateix concepte.
- L'equivalent als blocs d'MC en memòria virtual s'anomenen *pàgines*.
- Les fallades d'MV s'anomenen *fallades de pàgina*.

7.2.1 Unitat bàsica de gestió de memòria: la pàgina

- Una pàgina virtual és un bloc de memòria contigu i de grandària fixa T d'un programa.
- Per exemple $T = 4\text{KiB} = 2^{12}$ bytes.
- És la unitat més petita de gestió de memòria amb que treballa l'MV.

7.2.1 Unitat bàsica de gestió de memòria: la pàgina

- La subdivisió d'un programa en pàgines la fem pensant en l'espai d'adreçament lògic del programa.
- Cada pàgina tindrà un *número de pàgina virtual* (virtual page number o VPN).
- Donada una adreça lògica A , $VPN = \frac{A}{T} = 32 - t$ bits alts de l'adreça.
- *page offset* = t bits de menor pes de l'adreça.

7.2.1 Unitat bàsica de gestió de memòria: la pàgina

Exemple:

- $A = 0x10010004$ i $T = 4\text{KiB} = 2^{12}$.
- $\text{VPN} = 0x10010$.
- $\text{Page offset} = 0x004$.

7.2.1 Unitat bàsica de gestió de memòria: la pàgina

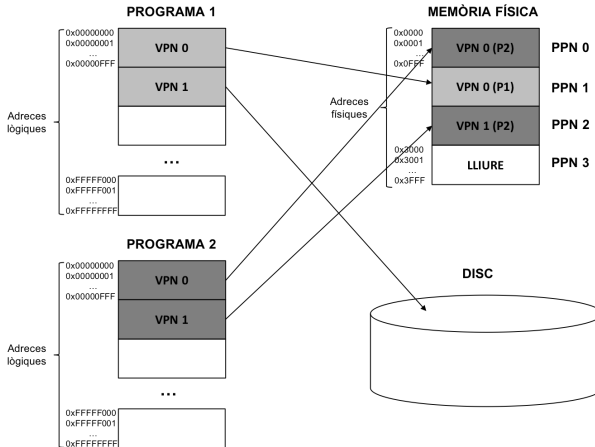
- Durant l'execució, caldrà anar carregant a l'MF les pàgines que es vagin necessitant.
- Cada subdivisió contigua de grandària T de la memòria física l'anomenarem *marc de pàgina* (page frame en anglès).
- Cada marc de pàgina tindrà associat un *número de pàgina física* (physical page number o PPN).
- Les pàgines d'un programa que, en un moment donat, no estiguin a la memòria física estaran al disc.

7.2.1 Unitat bàsica de gestió de memòria: la pàgina

Exemple:

- Adreces de 32 bits i una mida de pàgina $T = 4\text{KiB} = 2^{12}$ bytes.
- L'espai d'adreçament lògic de cada programa seria de 2^{32} bytes = $4\text{GiB} = 2^{32}/2^{12} = 2^{20}$ pàgines virtuals.
- Dos programes que fan servir 2 pàgines virtuals cadascú (VPN 0 i VPN 1).
- Memòria física de 2^{14} bytes = $16\text{KiB} = 2^{14}/2^{12} = 4$ marcs de pàgina (PPN 0, PPN 1, PPN 2 i PPN 3).

7.2.1 Unitat bàsica de gestió de memòria: la pàgina



7.2.1 Unitat bàsica de gestió de memòria: la pàgina

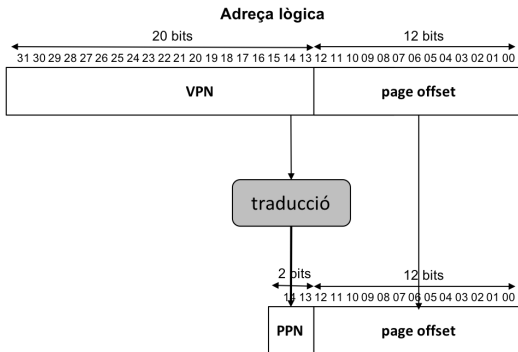
- L'assignació de pàgines a marcs de pàgina la realitza el sistema operatiu en funció de l'espai disponible en cada moment.
- **El VPN no determina on van les pàgines**, com passava en una memòria cache completament associativa.

7.2.2 Traducció d'adreces

- El processador treballa amb adreces lògiques.
- Cada vegada que necessita llegir o escriure una dada pregunta per l'adreça lògica d'aquesta dada a la *unitat de gestió de memòria* (MMU).
- *Traducció d'adreces*: la MMU tradueix l'adreça lògica en l'adreça física on realment es troba la dada.

7.2.2 Traducció d'adreces

- Donat el número de pàgina (el VPN) a la que pertany l'adreça, el sistema de traducció determina en quin marc de pàgina (PPN) de la memòria física es troba.



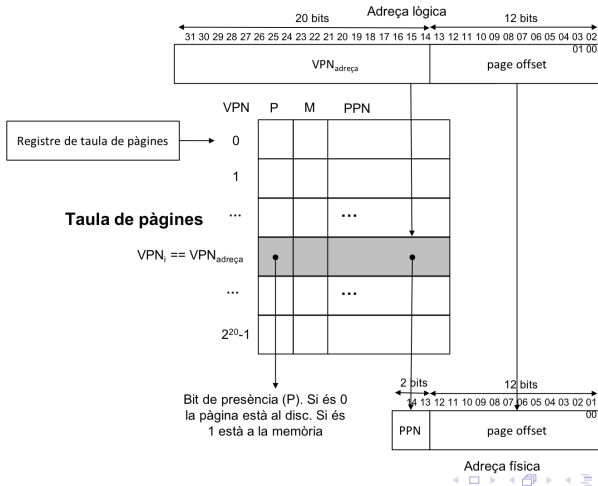
7.2.3 Localització fàcil amb taula de pàgines

- La selecció de a quin marc de pàgina es carrega una pàgina donada la realitza el sistema operatiu.
- Per poder recordar després (traducció d'adreces), a quin marc de pàgina s'ha carregat la pàgina amb un VPN determinat, es fa servir una taula anomenada *taula de pàgines*.
- Hi ha una taula de pàgines per cada programa.

7.2.3 Localització fàcil amb taula de pàgines

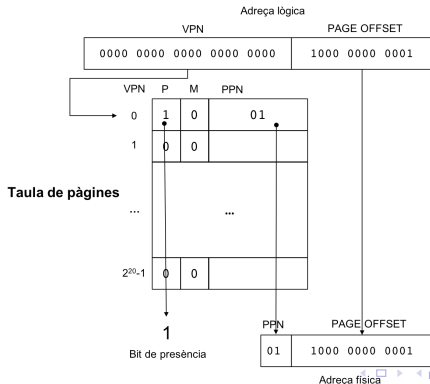
- Cada filera de la taula de pàgines s'anomena *entrada de la taula de pàgines* (PTE).
- Hi ha tantes entrades com pàgines virtuals hi hagi.
- A l'exemple anterior, la taula de pàgines de cadascun dels programes tindria 2^{20} entrades.

7.2.3 Localització fàcil amb taula de pàgines



7.2.3 Localització fàcil amb taula de pàgines

Exemple: traducció de l'adreça lògica 0x00000801 mitjançant la taula de pàgines corresponent al Programa 1 de l'exemple anterior.



7.2.3 Localització fàcil amb taula de pàgines

- La taula de pàgines la gestiona el sistema operatiu.
- Per simplicitat, suposem de moment que s'emmagatzema a la memòria.

7.2.3 Localització fàcil amb taula de pàgines

- Si hi ha múltiples programes executant-se concurrentment, cadascú disposarà de la seva pròpia taula de pàgines.
- Quan hi ha múltiples programes en execució es va repartint el control del CPU entre ells (un CPU).
- En un instant de temps, només hi ha un programa en execució, el que anomenen procés actiu.

7.2.3 Localització fàcil amb taula de pàgines

- Per saber on està la taula de pàgines del procés actiu, el hardware inclou un registre que apunta a la posició inicial de la seva taula de pàgines.
- El registre de la taula de pàgines, el PC i els registres de propòsit general determinen el que s'anomena *estat del procés actiu*.

Índex

- 1 7.1 Introducció
- 2 7.2 Funcionament de la memòria virtual
- 3 7.3 Fallada de pàgina**
- 4 7.4 Traducció ràpida amb TLB
- 5 7.5 Protecció i Compartició

7.3 Fallada de pàgina

- Es produeix una *fallada de pàgina* (page fault) quan la CPU referència una adreça lògica pertanyent a una pàgina que no es troba a la memòria física.
- És a dir, que el bit P de l'entrada de la TP corresponent al VPN sol·licitat val 0.
- Quan això succeeix cal:
 - Llegir del disc la pàgina.
 - Carregar-la en un marc de pàgina de la memòria física.
 - Actualitzar la informació de la TP.
 - Reintentar l'operació.

7.3 Fallada de pàgina

- Passem per alt, de moment, la manera que té el sistema operatiu de localitzar la pàgina al disc.
- Mai llegirem o escriurem dades individuals directament al disc, sempre carregarem primer la pàgina a l'MF.
- Per tant, pel que fa a les escriptures, podem dir que seguirem una política **d'escriptura diferida amb assignació**.

7.3.1 Reemplaçament d'una pàgina

- Si cal carregar una pàgina a MF i no queda cap marc de pàgina lliure? Reemplaçament.
- Algorisme de reemplaçament de pàgines, e.g. LRU.
- Com fem servir escriptura retardada cal un bit M de pàgina modificada"(bit M).
- Abans de reemplaçar una pàgina amb bit $M = 1$ cal escriure al disc la pàgina modificada.
- A la zona del disc on s'emmagatzemen les pàgines reemplaçades se l'anomena *espai d'intercanvi* (swap space).

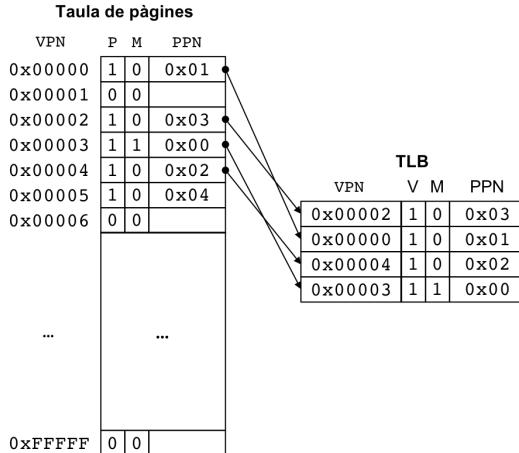
Índex

- 1 7.1 Introducció
- 2 7.2 Funcionament de la memòria virtual
- 3 7.3 Fallada de pàgina
- 4 7.4 Traducció ràpida amb TLB**
- 5 7.5 Protecció i Compartició

7.4.1 El TLB, una cache de traduccions

- Per llegir o escriure una dada caldrà accedir abans a la TP.
- Però TP a la memòria! Dos accessos a memòria cada vegada :-)
- TLB = cache de traduccions (translation-lookaside buffer).
- Emmagatzema les darreres entrades utilitzades de la TP.
- És un component hardware (part de la MMU).

7.4.1 El TLB, una cache de traduccions



7.4.1 El TLB, una cache de traduccions

- Entrada del TLB = VPN + còpia d'una entrada de la TP (bits P i M, i PPN).
- Com es fa la cerca? Per VPN.
- Però cal saber si l'entrada està inicialitzada: farem servir el mateix bit P.
- Per això l'anomenarem bit V (Validesa).

7.4.2 Encert de TLB

- Quan calgui traduir una adreça, primer es buscarà el VPN entre els VPNs del TLB.
- Si es troba (encara que V val 0): *encert de TLB* (TLB hit).
- Utilitzarem la informació per traduir.

7.4.2 Encert de TLB

- Si el bit V val 0, després de l'encert de TLB es produirà una fallada de pàgina.
- Un cop resolta (i actualitzada la TP) es reescriurà l'entrada al TLB.
- Finalment **reintentem tot el procés**

7.4.3 Fallada de TLB

- Si el VPN no es troba al TLB: *fallada de TLB* (TLB miss).
- Copiem entrada TP al TLB (sense mirar si P és 0 o 1).
 - Reemplacem primer entrades TLB amb el bit V a 0.
 - Si no n'hi ha cap: algorisme de reemplaçament aleatori.
- Finalment **reintentem tot el procés**.

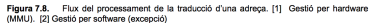
7.4.4 El bit V

- El bit V pot valer 0 per dos motius:
 - Entrada TLB no inicialitzada.
 - Entrada copiada de TP amb $P=0$.
- Però hi podria haver, per casualitat, encert de TLB en una entrada no inicialitzada!
- No importa, provocarà fallada de pàgina i reescriptura entrada TLB.

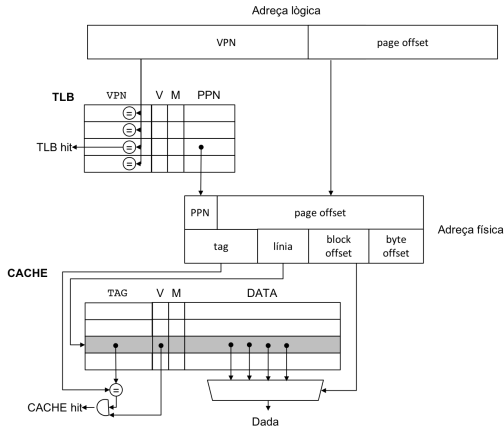
7.4.5 Gestió de les escriptures amb TLB

- Si encert de TLB ens estalviarem accedir a la TP per fer la traducció.
- Però, i si l'accés és una escriptura i el bit M al TLB val 0?
- Posarem simultàniament a 1 els bits M al TLB i a la taula de pàgines.
- Només ho farem quan el bit M valgui 0 (la primera vegada).

7.4 Traducció ràpida amb TLB



7.4.7 Integració de la memòria virtual i la memòria cache



7.4.8 Exemple pràctic

- Espai d'adreçament lògic de 2^{32} bytes = 4GiB = $2^{32}/2^{12} = 2^{20}$ pàgines virtuals.
- Memòria física de 2^{14} bytes = 16KiB = $2^{14}/2^{12} = 4$ marcs de pàgina.
- Un únic programa en execució (PPN 3 lliure).
- L'ordre en que s'han produït els accessos previs ha estat: VPN 3, VPN 0, VPN 2.
- Per claredat al TLB el reemplaçament serà LRU en comptes d'aleatori.

7.4.8 Exemple pràctic

Taula de pàgines (a l'inici)

VPN	P	M	PPN
0x00000	1	0	0x01
0x00001	0	0	
0x00002	1	0	0x02
0x00003	1	1	0x00
0x00004	0	0	
0x00005	0	0	
0x00006	0	0	
...	0	0	...
0xFFFFF	0	0	

TLB (a l'inici)

VPN	V	M	PPN
0x00003	1	1	0x00
0x00000	1	0	0x01
0x00002	1	0	0x02
0xAFB3	0	0	0xFA

Seqüència d'accessos a memòria

adreça	VPN	TLB miss	VPN _{TLB_OUT}	page fault	esc. disc	lect. disc	PPN
E:0x00002A0B	0x00002	No	-	No	-	-	0x02
L:0x00001F21	0x00001	Si	-	Si	-	0x00001	0x03
L:0x0000420C	0x00004	Si	0x00003	Si	0x00003	0x00004	0x00
L:0x00003001	0x00003	Si	0x00000	Si	-	0x00003	0x01
L:0x00005120	0x00005	Si	0x00002	Si	0x00002	0x00005	0x02

Índex

- 1 7.1 Introducció
- 2 7.2 Funcionament de la memòria virtual
- 3 7.3 Fallada de pàgina
- 4 7.4 Traducció ràpida amb TLB
- 5 7.5 Protecció i Compartició

7.5.1 Protecció

- L'MV permet compartir la memòria del computador de manera segura entre múltiples processos.
- Un procés no ha de poder accedir a l'espai d'adreçament d'un altre procés o del sistema operatiu.
- La traducció via TP ho garanteix si assumim que els processos no comparteixen cap pàgina física.

7.5.1 Protecció

- Podria un procés modificar la seva pròpia TP?
- No, les TPs van a l'espai d'adreçament reservat al S.O.
- El S.O. no és un procés, sinó un programari comú a tots els processos.
- El S.O. té reservada una part de l'espai d'adreçament de tots els processos.
- en MIPS, les adreces lògiques amb el bit 31=1.

7.5.1 Protecció

- El processador disposa de dos modes de funcionament, mode usuari i mode sistema.
- Només en mode sistema serà possible modificar el TLB o les TPs.

7.5.1 Protecció

Protecció contra escriptura:

- Es pot prohibir l'escriptura en determinades pàgines i permetre-ho en altres.
- El motiu el veurem a la següent subsecció (compartició).
- Bit de permís d'escriptura (E) que s'inclou en cada entrada de la TP i del TLB.

7.5.2 Compartició

- Un procés P1 vol permetre a un altre, P2, accedir al seu espai d'adreçament.
- El sistema operatiu, a petició de P1, assigna una pàgina lògica de P2 a la mateixa pàgina física que P1 vol compartir.
- La pàgina compartida pot tenir permís d'escriptura o no per al procés P2.

7.5.2 Compartició

