

Tema 7: Memòria Virtual

Professor: Francesc Solsona

Departament d'Informàtica i Enginyeria Industrial
Escola Politècnica Superior
Universitat de Lleida

9 de desembre de 2025

Continguts

- 1 Introducció
 - Paginació sota demanda i Precàrrega
 - Procediment de Fallada de Pàgina
- 2 Temps Efectiu d'Accés (TEA) a MP
- 3 Algorismes d'Administració de Mem. Virtual
 - Algorismes d'Assignació (de cel·les)
 - Algorismes de Reemplaçament (de pàgines)
- 4 Hiperpaginació
 - Model del Conjunt de Treball
 - Model de Freqüència de Fallades de Pàgina

Introducció

Def. Memòria Virtual: Transferència de pàgines entre Memòria Principal (MP) i Memòria Secundària.

Avantatges:

- Augment del grau de multiprogramació.
- Permet executar programes que no caben en MP.

Inconvenients:

- Execució més lenta dels programes.

Existeix dues tècniques que implementen la Memòria Virtual:

Paginació sota demanda (més utilitzada) i **Precàrrega**.

Paginació sota demanda i Precàrrega

Def. Paginació sota demanda: una pàgina no es carrega a MP fins que no s'hi accedeix. A mesura que es van produint fallades, es van carregant les pàgines de Memòria Secundària a MP. Inicialment, quan comença l'execució d'un procés, en MP no hi ha cap pàgina carregada del procés en qüestió.

Def. Precàrrega: consisteix en carregar pàgines en MP per anticipat (no per demanda). Quan es produeix una fallada de pàgina, es porten a més de la pàgina que ha fallat, altres pàgines que es consideri que necessitarà el procés. P.e., portar 2 pàgines consecutives més. Beneficiosa depenent de si hi ha encert en la predicció.

Procediment de Fallada de Pàgina

Procediment per tractar una *fallada de pàgina* (s'ha accedit per exemple, a una pàgina "F" que no està carregada en MP):

- ❶ Si (cel·les lliures) Lectura de pàgina F (Pàgina que ha fallat) en cel·la lliure
- ❷ Sinó
 - ❶ Selecció de la Pàgina Víctima V (en cel·la C) mitjançant un Algorisme de Reemplaç
 - ❷ Marcar V com invàlida (dins la Taula de Pàgines)
 - ❸ Si V modificada (bit *Modificat* de V actiu)
V s'escriu en Memòria Secundària
 - ❹ Lectura de pàgina F (Pàgina que ha fallat) en cel·la C
 - ❺ Marcar F com pàgina vàlida (bit de *Validesa* actiu)
- Mentre es fan aquestes tasques es pot despatxar un nou procés

Temps Efectiu d'Accés (TEA) a MP (1/3)

Def. TEA: temps mig d'accés a Memòria Principal. S'ha de tenir en compte temps i percentatges d'accés a MP que produeixen (i que no produeixen) fallades de pàgina.
Formalment:

$$TEA = [(1 - P) * (T_a)] + [P * (T_{fp})],$$

on P = Probabilitat fallada de pàgina; T_a = Temps d'accés a MP;
 T_{fp} = Temps d'accés a MP tenint en compte que s'ha produït una fallada de pàgina.

Si el dispositiu de paginació és el disc, cal definir els temps següents:

Temps de cerca (T_c): temps mig en cercar la pista/cilindre

Temps de latència (T_l): temps mig en cercar el sector dins la pista

Temps de transferència (T_t): temps en transferir una pàgina de MP (disc) a disc (MP)

Temps Efectiu d'Accés (TEA) a MP (2/3)

Exemple: disposem d'un disc que gira a 7.500 rev/min. $T_c = 2\text{ms}$ i transfereix 100.000 paraules/s. $P=0,25$ i la mida d'una pàgina és 1.000 paraules. El sistema de gestió de memòria és Paginació (la taula de pàgines s'implementa en MP) i el Temps d'accés (T_a) a MP és $4\mu\text{s}$. Calcular TEA.

$$T_a = 4 * 10^{-6} \text{s}, \quad T_c = 2 * 10^{-3} \text{s},$$

$$T_l \Rightarrow 7.500 \frac{\text{rev.}}{\text{min.}} * \frac{1 \text{ min.}}{60 \text{ seg.}} = 125 \frac{\text{rev.}}{\text{seg.}} \Rightarrow T_l = \frac{1}{2} * \frac{1}{125} \text{s} = 4 * 10^{-3} \text{s},$$

$$T_t \Rightarrow 100.000 \frac{\text{paraules}}{\text{segon}} * \frac{1 \text{ pàgina}}{1.000 \text{ paraules}} = 100 \frac{\text{pàgines}}{\text{segon}} \Rightarrow T_t = 10^{-2} \text{s} = 10 * 10^{-3} \text{s}$$

$$T_{MP-MS} = T_c + T_l + T_t = (2 * 10^{-3} + 4 * 10^{-3} + 10 * 10^{-3}) \text{s} = 16 * 10^{-3} \text{s}.$$

Temps Efectiu d'Accés (TEA) a MP (3/3)

$$TEA = [(1 - P) * (2 * T_a)] + [P * (T_{MP-MS} + 3 * T_a)] s$$

$$TEA = [(1 - 0,25) * (2 * 4 * 10^{-6})] + [0,25 * (16 * 10^{-3} + 3 * 4 * 10^{-6})] s$$

$$TEA = [(0,75) * (0,008 * 10^{-3})] + [0,25 * (16 * 10^{-3} + 0,012 * 10^{-3})] s$$

$$TEA = (0,75) * (0,008 * 10^{-3}) + 0,25 * (16,012 * 10^{-3}) s$$

$$TEA = 0,006 * 10^{-3} + 4,003 * 10^{-3} s$$

$$TEA = 4,009 ms$$

Algorismes d'Administració de MV

Hi ha 2 tipus d'algorismes d'administració de la Mem. Virtual:

- ❶ **Algorismes d'assignació** (de cel·les): tipus d'algorisme que assignen cel·les als processos. Tipus d'algorismes:
 - ❶ **Assignació Local.**
 - ❷ **Assignació Global.**
- ❷ **Algorismes de reemplaçament** (de pàgines): tipus d'algorisme que tria la pàgina de MP a reemplaçar (anomenada pàgina víctima) en cas de fallada de pàgina i no haver-hi cap cel·la lliure. Tipus d'algorismes:
 - ❶ Òptim
 - ❷ FIFO
 - ❸ Rel·lotge (o segona oportunitat)
 - ❹ LRU
 - ❺ Buffering de pàgines

Assignació Local

Assignació Local: l'algorisme de reemplaç sol pot reemplaçar pàgines del procés que ha causat la fallada. En aquest cas cal saber el **mínim nombre de cel·les que cal assignar a un procés**.

- Aquest **mínim** és igual al **màxim nombre de cel·les accedides en l'execució d'una instrucció**. És a dir, depèn del format d'una instrucció (nombre de paraules que ocupa) i del tipus d'adreçament tant dels operands com del resultat (immediat: 0, directe: 1, indirecte: 2, indexat: 1). Ho veurem per mitjà d'un exemple.

Assignació Local

exemple: suposeu que el format d'una instrucció ocupa dos paraules (mida paraula = 1 Byte):

OPCODE	OP ₁
OP ₂	RES

Mida instrucció:	2 paraules	(2 cel·les)
OP ₁ (Operand 1):	adreçament indirecte	(2 cel·les)
OP ₂ (Operand 2):	adreçament indirecte	(2 cel·les)
RES (Resultat):	adreçament directe	(1 cel·la)
Total :		<hr/> 7 cel·les

Assignació Global

Assignació Global: l'algorisme de reemplaç pot reemplaçar qualsevol pàgina de qualsevol procés.

- Hi ha 2 algorismes (serveixen també per l'assignació local):
 - 1 Algorisme d'**Assignació Igualitària**
 - 2 Algorisme d'**Assignació Proporcional**

Algorisme d'assignació Igualitària

Suposeu el següent:

- Mida de Memòria Principal: M cel·les
- N processos

⇒ S'assigna $\lfloor M/N \rfloor$ cel·les a cada procés. Les cel·les no assignades aniran a la llista de cel·les lliures. Com a molt hi haurà $M \bmod N$ cel·les a la llista de cel·les lliures.

Algorisme d'assignació Proporcional

Suposeu el següent:

- Mida de Memòria Principal: M cel·les
- N processos
- S_i : mida Memòria virtual requerida pel procés P_i
- $S = \sum S_i$

⇒ S'assigna $\lfloor (S_i/S) * M \rfloor$ cel·les a cada procés. En aquest cas, com a molt aniran N cel·les a la llista de cel·les lliures.

Algorismes de Reemplaçament (Introducció)

- Def. **Retenció de pàgines en Memòria**: pàgines marcades com no reemplaçables. En Linux, existeix la crida a sistema *mlock* per aquest fi:

```
#include <sys/mman.h>  
  
int mlock(const void *addr, size_t len);
```

desactiva el paginat per les pàgines de memòria en el grup d'adreces que comença en *addr* i amb una mida *len* Bytes.
- Def. **Anomalia de Belady**: donat un algorisme de reemplaç, en augmentar el número de cel·les, augmenta també el número de fallades de pàgina.

Algorisme Òptim

Pàgina víctima \Rightarrow pàgina resident que tardarà més temps en ser referenciada. No es pot realitzar. Es poden fer aproximacions sabent la història passada de referències de pàgines.

Exemple: Seqüència de referències de pàgines: 0, 7, 5, 8, 10, 12
..., 0,..., 10, ... 8, 5, 9, 7

0	0	0	0
7	8	8	12
5	5	10	10

Algorisme FIFO

Pàgina víctima \Rightarrow pàgina que porta més temps resident.

Exemple: Seqüència de referències de pàgines: 0, 7, 5, 8, 10, 12

...

0	7	5	8	10	12
	0	7	5	8	10
		0	7	5	8

Algorisme del rellotge (o 2^a oportunitat)

Idea \Rightarrow FIFO + ús del bit de *Referència* (indica pàgina Referenciada/accedida). Es pot implementar mitjançant una llista circular.

Funcionament:

- ❶ Si pàgina escollida per FIFO no té actiu el bit de *Referència* s'escull com a pàgina víctima
- ❷ Si té el bit de *Referència* actiu (2^a oportunitat) es desactiva el bit de *Referència* es continua per la següent pàgina

Algorisme LRU (Least Recently Used)

Pàgina víctima \Rightarrow pàgina resident menys recentment usada

Implementació:

- En cada entrada de la Taula de Pàgines hi ha un nou camp (temps)
- En cada accés a MP es copia el temps actual del sistema en el camp temps de l'entrada referenciada
- Pàgina víctima = pàgina amb camp temps més baix

Exemple: Seqüència de referències de pàgines: 0, 7, 5, 8, 7, 10, 9, 10

0	8	8	9
7	7	7	7
5	5	10	10

Buffering de pàgines

Idea \Rightarrow mantenir una reserva de cel·les lliures. Quan es produeix una fallada de pàgina es tria una cel·la d'una llista de cel·les lliures. A més es segueix l'algorisme següent:

cada període de temps T

Mentre (cel·les lliures $<$ límit)

moure pàgines no referenciades més antigues a la llista de cel·les lliures

Si es referencia una pàgina mentre està en la llista de cel·les lliures es recupera la pàgina de la llista de cel·les lliures

Hiperpaginació

Def. Hiperpaginació: taxa excessiva de fallades de pàgina d'un procés o en el sistema (veure Figura). El grau d'utilització de la CPU cau degut a que els processos estan gairebé sempre paginant.

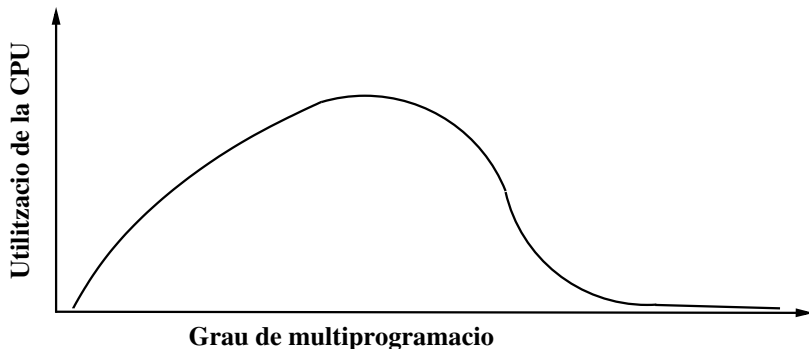


Figura: Hiperpaginació

Model del Conjunt de Treball

Def. Localitat: conjunt de pàgines utilitzades en un determinat bloc d'un programa. P.e.: conjunt de pàgines d'una funció.

Def. Conjunt de Treball (d'un procés): pàgines utilitzades pel procés en les últimes Δ referències més recents.

- Si
 Δ petit: aproximació dolenta a la localitat
 Δ gran: es pot solapar vàries localitats
- Amb assignació local: Hiperpaginació en P_i si
 $\#[\text{cel·les assignades a } P_i] < \text{Mínim } \#[\text{cel·les per procés}]$
- Amb assignació global: Hiperpaginació en el sistema si
 $\#[\text{cel·les en MP}] < \sum_i \#[\text{Conjunt de Treball de } P_i]$

Model del Conjunt de Treball

- Funcionament:

Assignació local:

Si $\#[\text{cel·les assignades a } P_i] < \text{Mínim } \#[\text{cel·les per procés}]$
augmentar $\#[\text{cel·les assignades a } P_i]$

Assignació global:

Si $\#[\text{cel·les en MP}] < \sum_i \#[\text{Conjunt de Treball de } P_i]$
es suspenen (o s'eliminen) processos

Si $\#[\text{cel·les en MP}] > \sum_i \#[\text{Conjunt de Treball de } P_i]$
s'augmenta el Grau de Multiprogramació

Problema del Model del Conjunt de Treball: difícil implementació.

Model de Freqüència de Fallades de Pàgina

- Aproximació del model del Conjunt de treball
- Funcionament:
 - Si taxa fallades de pàgina $<$ límit inferior \Rightarrow augmentar grau multiprogramació
 - Si taxa fallades de pàgina $>$ límit superior \Rightarrow es suspensen processos