

Fonaments dels Sistemes Operatius (FSO)

Departament d'Informàtica de Sistemes i Computadores (DISCA)
Universitat Politècnica de València

Bloc Temàtic 4: Gestió de Memòria

Unitat Temàtica 12 Memòria Virtual (II)

fSO

DISCA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

- **Objectius**

- Estudiar l'algorisme de reemplaçament de pàgines de **2ª oportunitat** com a aproximació al LRU
- Conèixer la problemàtica de la **hiperpaginació** i les solucions a la mateixa
- Analitzar les tècniques de **gestió de marcs** de memòria

- **Bibliografia**

- A. Silberschatz, P. B. Galvin. “Sistemas Operativos”. 7ª ed. Capítulo 9

- **Algorisme de reemplaçament de 2a oportunitat**
- Assignació de marcs
- Hiperpaginació
- Reserva de marcs

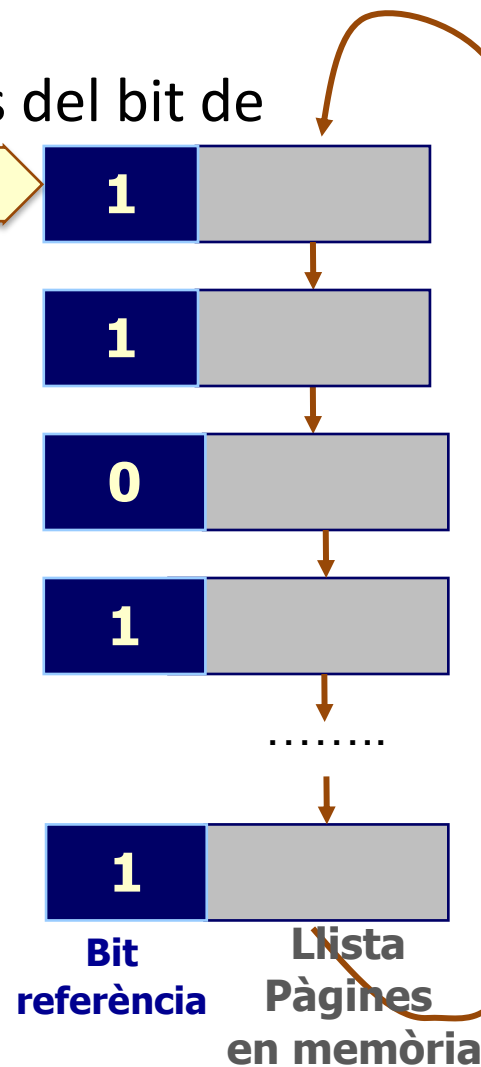
- Suportar un algorisme de reemplaçament LRU és car i ineficient
- Solució: Aproximació a LRU mitjançant l'ús del bit de referència

Algorisme de Segona Oportunitat

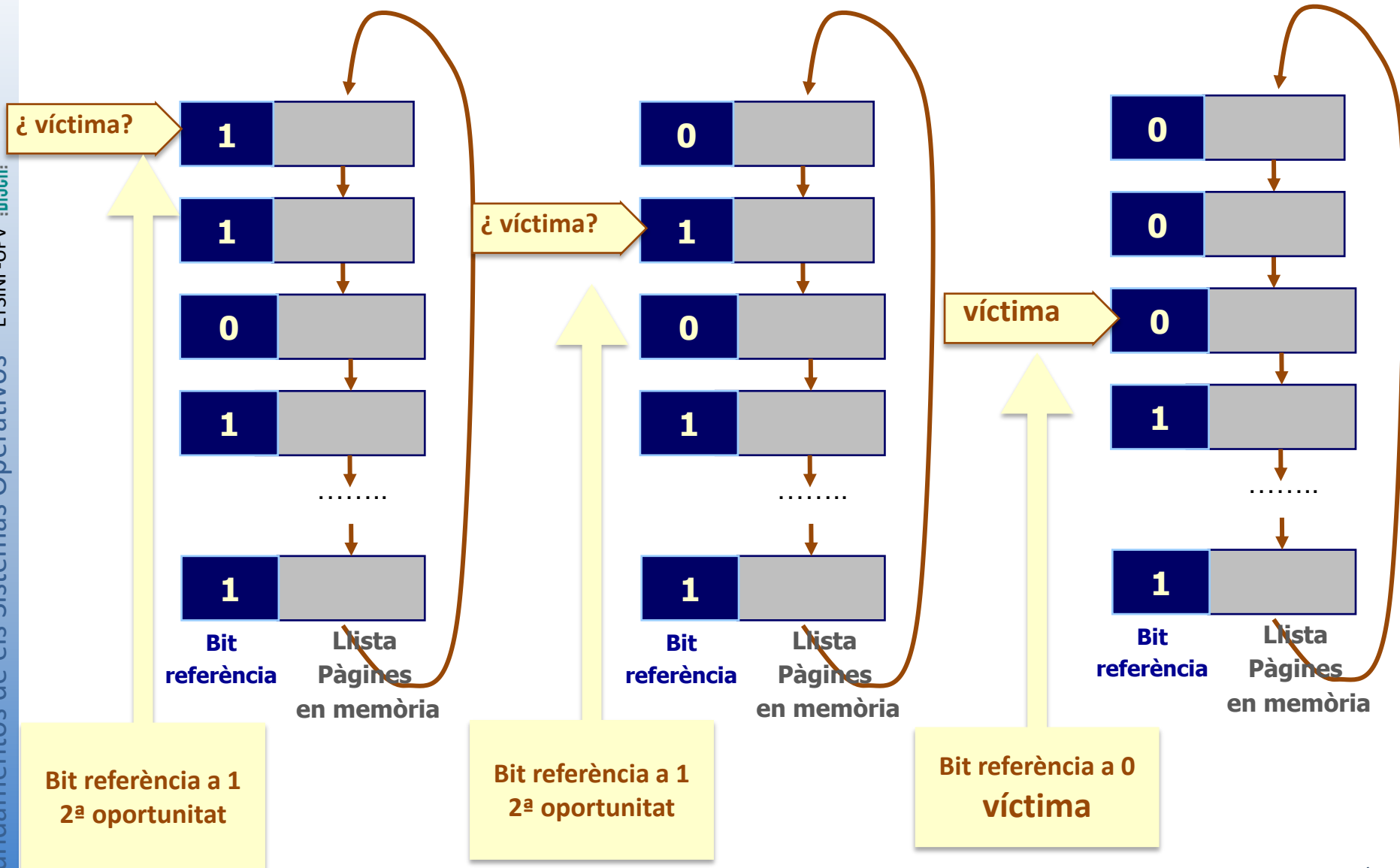
- Revisa el bit de referència de les pàgines ordenades amb criteri FIFO
- **Si el bit de referència de pàgina val 1**
 - Ficar el bit de referència a 0
 - Deixar la pàgina en memòria
- **Si el bit de referència és 0**
 - Es tria la pàgina com víctima.

¿ següent víctima?

2^a
oportunitat



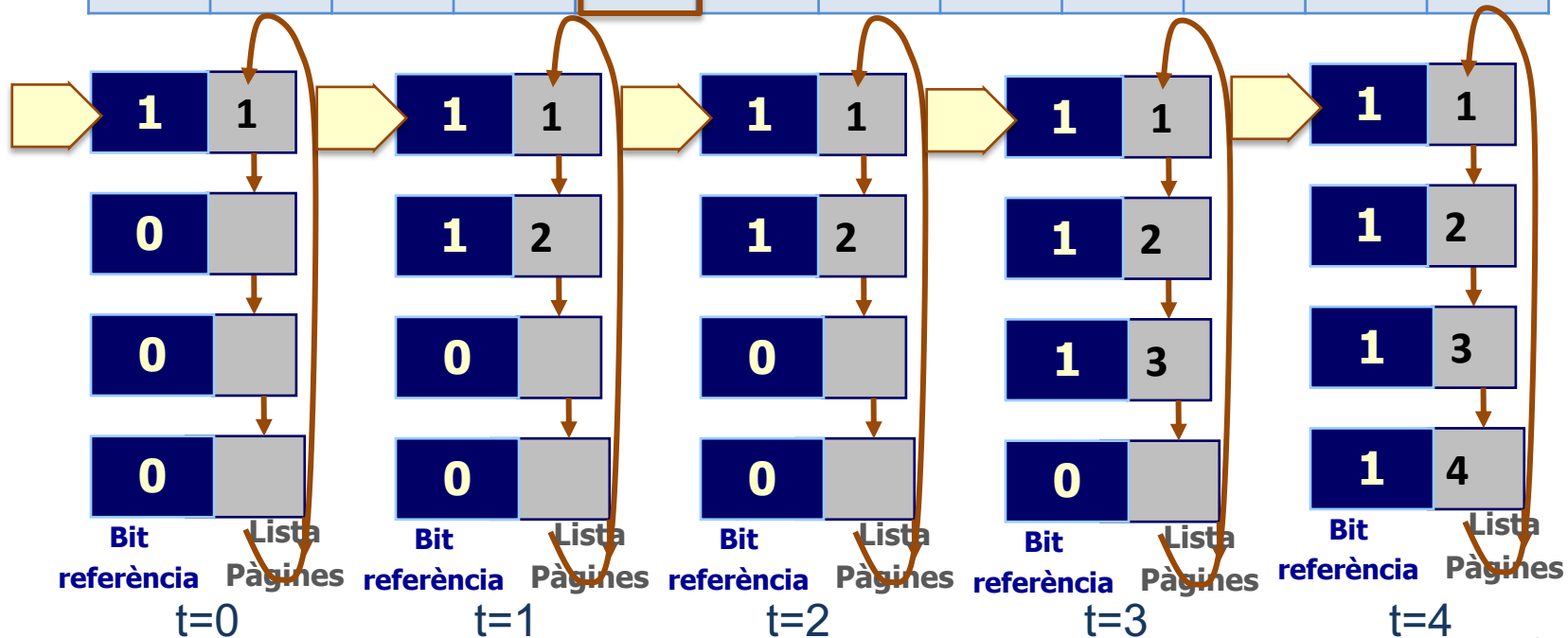
Aproximació a LRU mitjançant l'ús del bit de referència



Exemple: Memòria Principal de 4 marcs

T	0	1	2	3	4	5	6	7	8	9	10
Sèrie de referències	1	2	3	4	1	2	5	2	3	1	2

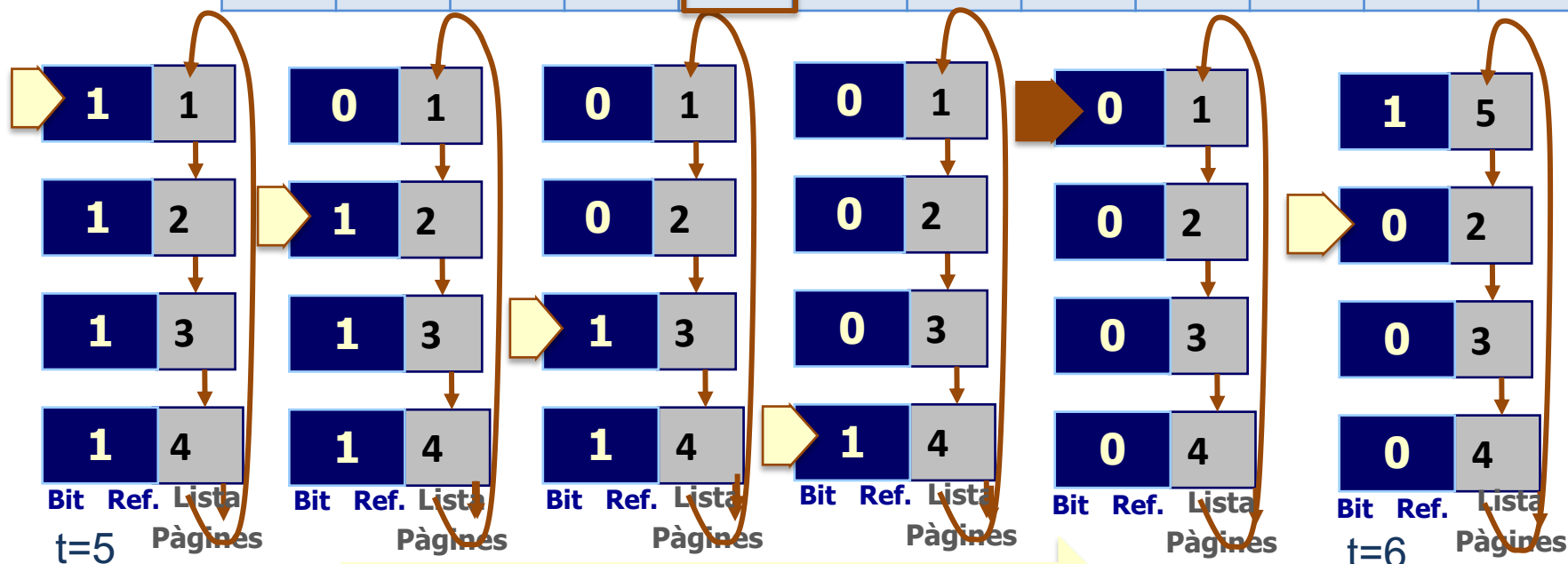
marc 0	1	1	1	1	1						
marc 1		2	2	2	2						
marc 2			3	3	3						
marc 3				4	4						



Exemple: Memòria Principal de 4 marcs

T	0	1	2	3	4	5	6	7	8	9	10
Sèrie de referències	1	2	3	4	1	2	5	2	3	1	2

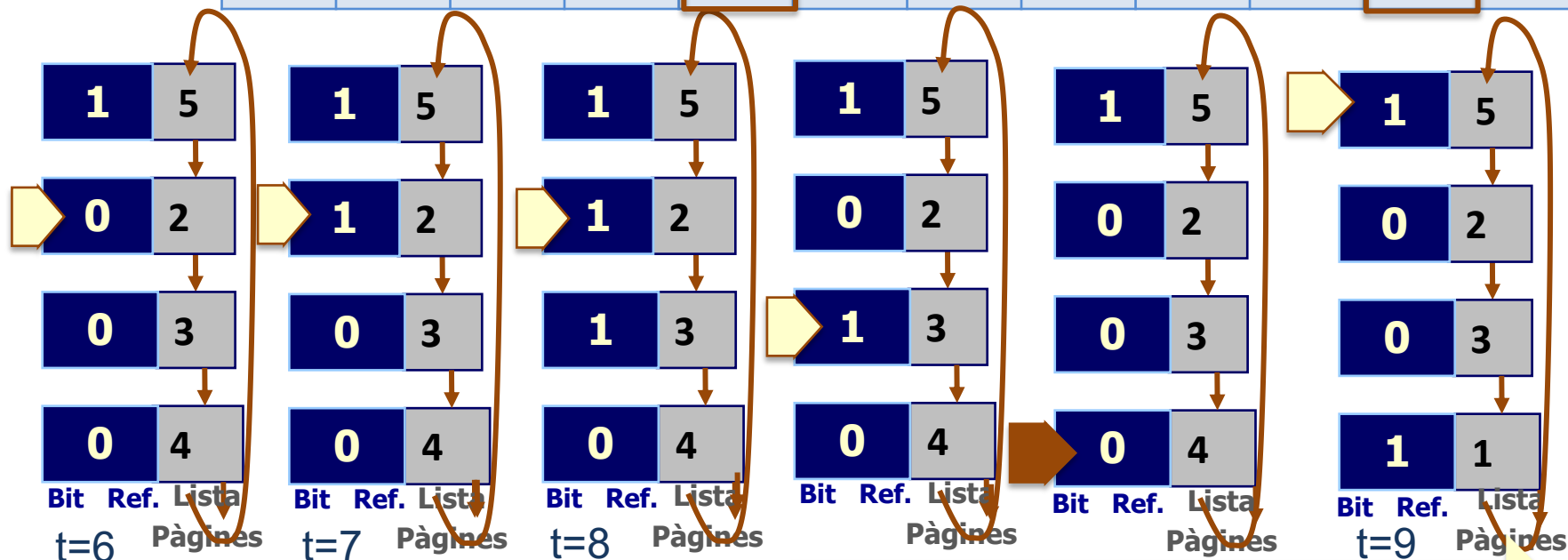
marc 0		1	1	1	1	1	1	5			
marc 1			2	2	2	2	2	2			
marc 2				3	3	3	3	3			
marc 3					4	4	4	4			



Cerquem victima amb segona oportunitat

Exemple: Memòria principal de 4 marcs

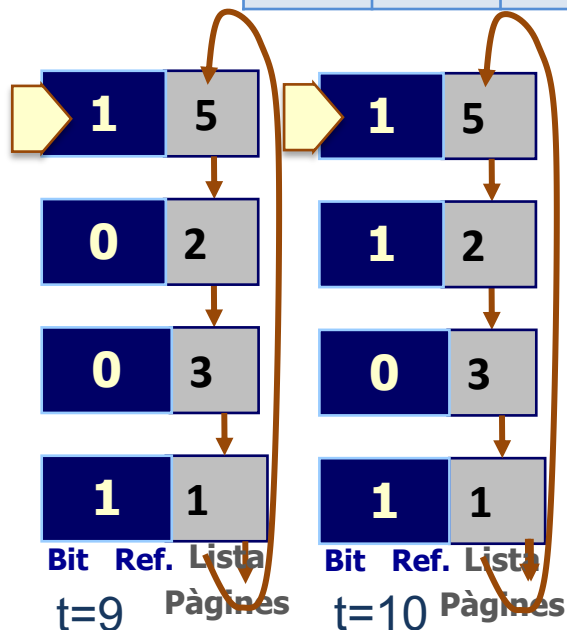
T	0	1	2	3	4	5	6	7	8	9	10
Sèrie de referències	1	2	3	4	1	2	5	2	3	1	2
marc 0	1	1	1	1	1	1	5	5	5	5	
marc 1		2	2	2	2	2	2	2	2	2	
marc 2			3	3	3	3	3	3	3	3	
marc 3				4	4	4	4	4	4	1	



Cerquem víctima amb segona oportunitat

Exemple: Memòria principal de 4 marcs

T	0	1	2	3	4	5	6	7	8	9	10
Sèrie de referències	1	2	3	4	1	2	5	2	3	1	2
marc 0		1	1	1	1	1	5	5	5	5	5
marc 1			2	2	2	2	2	2	2	2	2
marc 2				3	3	3	3	3	3	3	3
marc 3					4	4	4	4	4	1	1



Total 6 fallades de pàgina (2 amb reemplaçament)

- Algorisme de reemplaçament de 2a. oportunitat
- **Assignació de marcs**
- Hiperpaginació
- Reserva de marcs

- Problema d'assignació de marcs
 - Llista de marcs lliures:
 - La gestió de marcs requereix una **estructura de dades** on es mantinguen els **marcs lliures**
 - Política de repartiment de marcs entre els processos i el SO
 - Es dota al SO del nombre de marcs necessari per a executar-se
 - Es dota als processos d'un nombre mínim inicial de marcs i la resta sota demanda
 - El nombre mínim de marcs que ha d'assignar-se depèn de l'arquitectura del processador (nivell d'indirecció en el joc d'instruccions)

- Polítiques de **repartiment de marcs**

- **Assignació equitativa:** Donats **m marcs** i **n processos**, a tots els processos se li assignen **A_i marcs** per igual

$$A_i = m/n$$

- **Assignació proporcional:** En un sistema amb **n processos** $P_1 \dots P_i \dots P_n$, cadascún de grandària $S_1 \dots S_i \dots S_n$, a un procés P_i de grandària S_i se li assignen **A_i marcs** calculats com

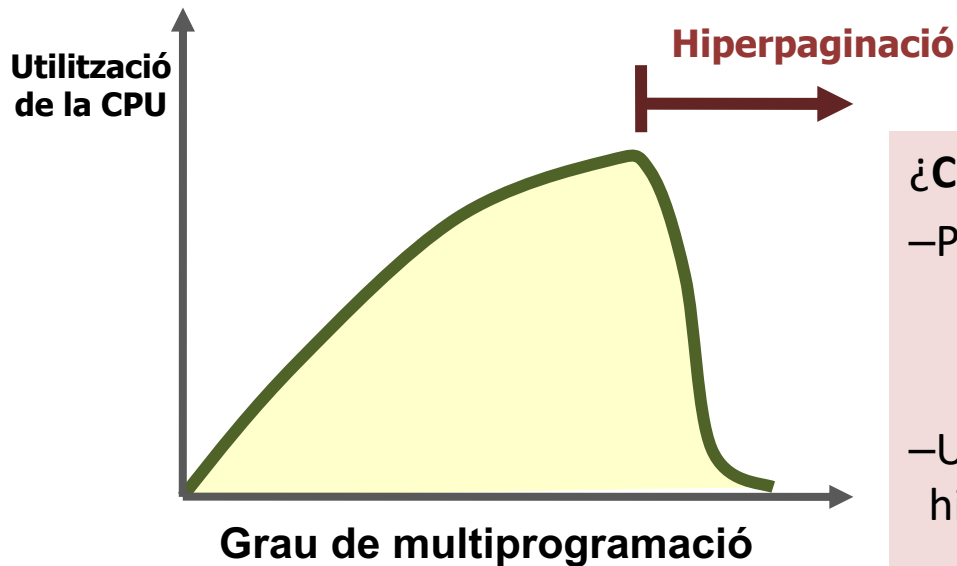
$$A_i = \frac{S_i * m}{\sum S_i}$$

- **Assignació prioritària:** s'assignen més marcs als processos més prioritaris

- **Àmbit de les polítiques de reemplaçament**
 - **Reemplaçament local:** el sistema només tria víctimes entre aquelles que ocupen marcs pertanyents al mateix procés que genera la fallada de pàgina
 - No es pot triar com víctima marcs d'altre procés
 - El nombre total de marcs assignats a un procés no varia
 - La seua execució no es veu influenciada per la de la resta de processos
 - Avantatges: Temps de resposta raonable
 - Inconvenient: Pitjor gestió global de memòria
 - **Reemplaçament global:** El sistema tria pàgines víctima d'entre totes les que tenen assignats marcs en memòria
 - Es poden triar com víctima marcs d'altres processos
 - El nombre total de marcs assignats a un procés pot variar
 - Avantatges: Gestió global de memòria optimitzada
 - Inconvenient: Temps de resposta elevat

- Algorisme de reemplaçament de 2a. oportunitat
- Assignació de marcs
- **Hiperpaginació**
- Reserva de marcs

- El problema de la **hiperpaginació**:
 - La memòria comença a escassejar i els processos generen **moltes fallades de pàgina** → passen més temps fent E/S que càlcul
 - El SO, front el **baix ús de la CPU**, accepta mes processos per a execució
 - Això agreuja el problema en repartir la mateixa memòria entre més processos → augmenta la taxa de fallades de pàgina



¿Com resoldre la hiperpaginació?

- Previndre i anticipar-se al problema
 - Utilitzar un model d'àrea activa
 - Control de la taxa de fallades
- Una vegada detectada la hiperpaginació
 - Passar processos a disc utilitzant planificador a mitjà termini

- Principi de **localitat de referència**

- Comportament dels programes segons el qual, basant-nos en el passat recent d'accessos, podem predir amb una precisió raonable quines instruccions i dades es referenciarà en un futur pròxim, ja que aquests no varien de forma abrupta sinó gradualment
- Localitat:
 - Conjunt de pàgines que un procés utilitza conjuntament.
 - Determinar-la es costós (requereix molts recursos)
- Es produeix hiperpaginació quan
$$\Sigma(\text{grandària de localitat}) > \Sigma(\text{grandària memòria total})$$

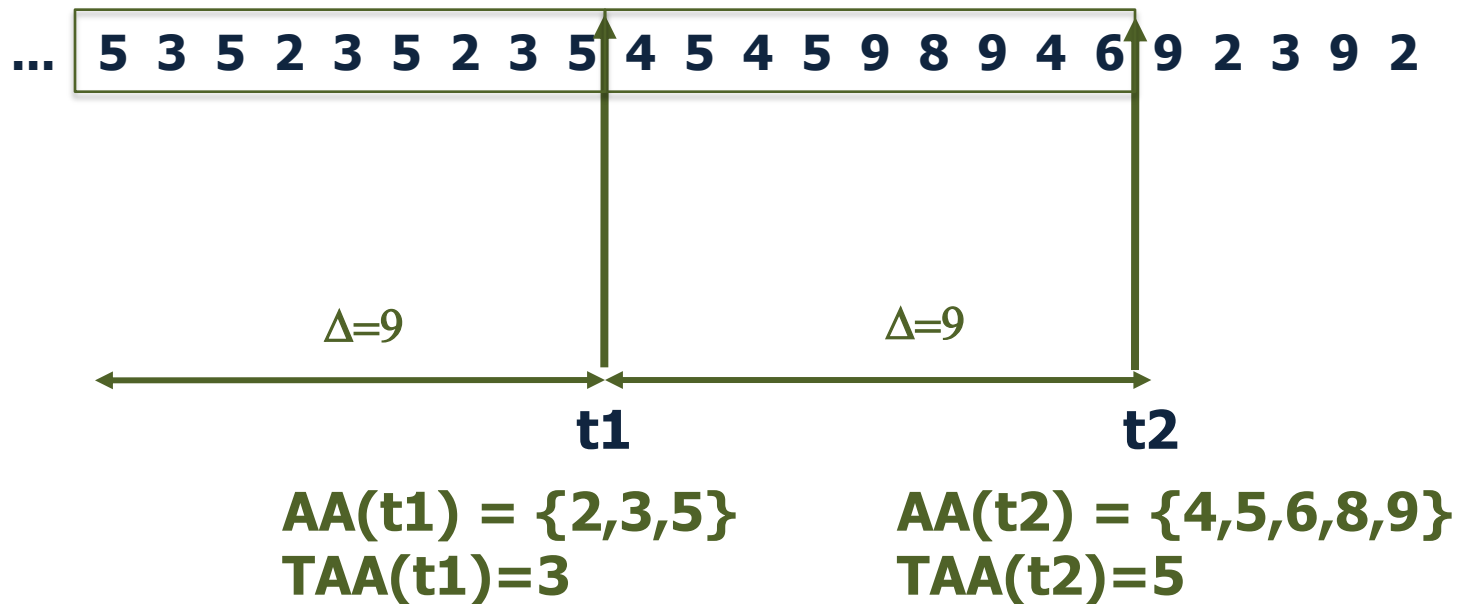
- **Model de l'àrea activa:** tècnica preventiva
 - Assumeix el principi de **localitat de referència**
 - Determinar el **nombre de pàgines que un procés ha de tindre en memòria simultàniament** per a obtenir un bon rendiment i evitar la hiperpaginació
 - **Àrea activa (AA) o “working set (WS)”**:
 - Conjunt de pàgines accedides en les últimes referències
 - Finestra d'àrea activa
 - Nombre fix Δ de referències utilitzades per a calcular el AA
 - AA format per les pàgines accedides en les Δ últimes referències
 - Grandària d'àrea activa: Nombre de pàgines diferents que conformen el AA_i del procés $P_i \rightarrow TAA_i$
 - En un sistema amb **m marcs** i **n processos** $P_1 \dots P_i \dots P_n$ es produeix **hiperpaginació** quan $\sum TAA_i > m$

- **Model de l'àrea activa**

- Exemple

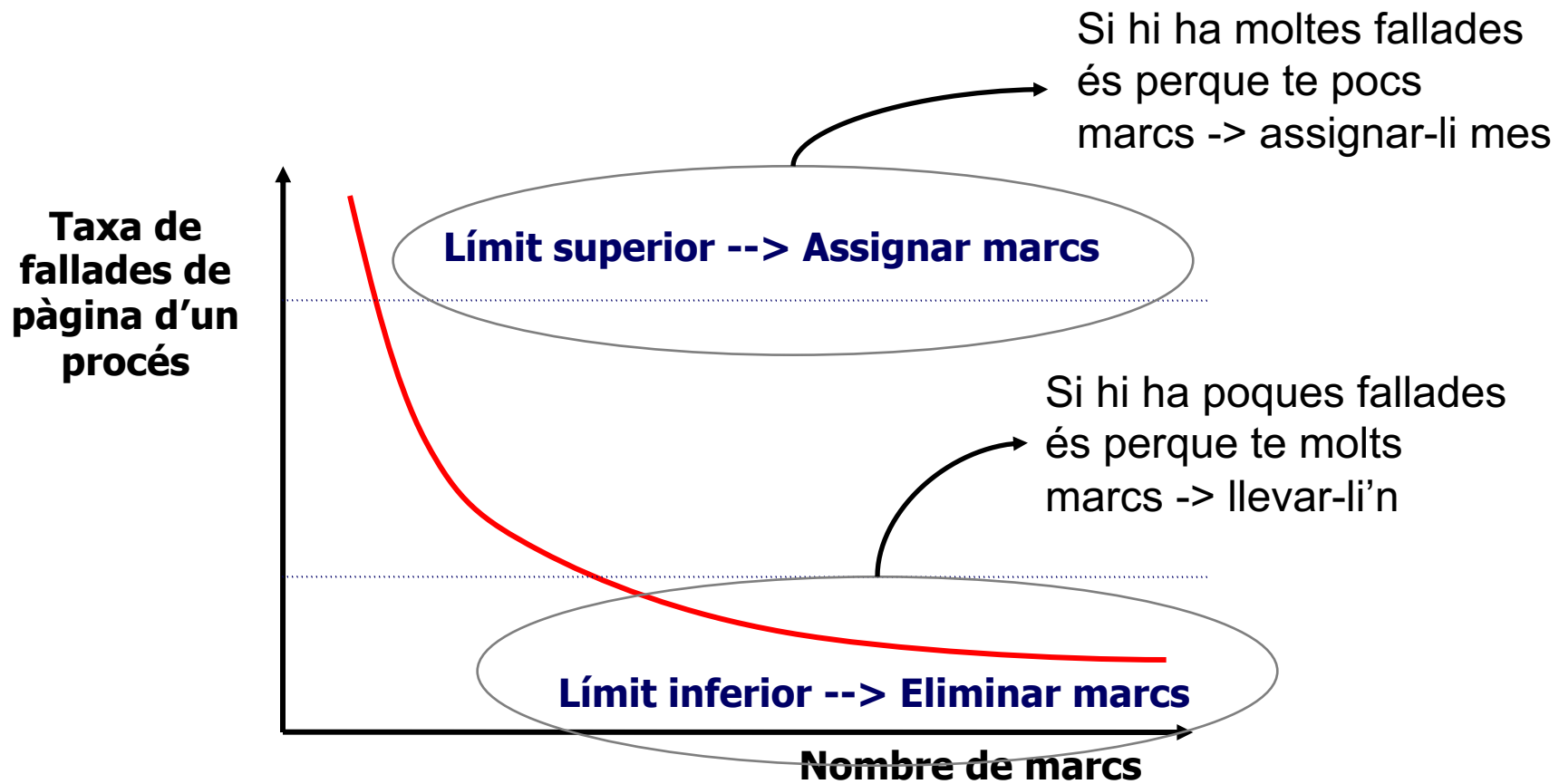
- Finestra d'àrea activa $\Delta = 9$
- Càlcul de AA per a un procés en els instants t1 i t2
- Càlcul de TAA

Sèrie de referències



- **Control de la taxa de fallades de pàgina**

- Tècnica preventiva que analitza directament la taxa de fallades per a determinar si s'està entrant en hiperpaginació



- Algorisme de reemplaçament de 2a. oportunitat
- Assignació de marcs
- Hiperpaginació
- **Reserva de marcs**

- **Concepte**

- Molts SO moderns guarden sempre un percentatge de la memòria principal com magatzem de marcs lliures. Aquest conjunt de marcs es coneix com **RESERVA DE MARCS**

- **Objectius**

- **Reduir el temps invertit per a servir una fallada de pàgina**
 - S'intenta tindre marcs lliures disponibles
 - S'hi aplica l'algorisme de reemplaçament
 - Només quan el nivell de marcs lliures baixa massa
 - Per a cercar diverses víctimes amortitzant el seu us
 - Realitzar el “page out”
 - S'escriuen diverses pàgines cada vegada en el disc
- Utilitzar la reserva per a **resoldre el problema d'hiperpaginació**

- **Gestió de la reserva de marcs**

- El S.O. garanteix que sempre hi haurà un cert nombre de **marcs lliures**
- Es fixen certs llindars:
 - Nombre mínim de marcs disponibles (MIN)
 - Nombre recomanable de marcs disponibles (REC)

REC >> MIN
- No cal utilitzar algorismes de reemplaçament molt eficients
 - Els primers sistemes VMS utilitzaven l'algorisme FIFO, ja que la seua MMU no tenia bit de referència.
 - És normal utilitzar un algorisme de segona oportunitat (Windows, UNIX SVR4, UNIX 4.4BSD, Linux, HP OpenVMS, ...).

- **Gestió de reserva de marcs: Procés monitor**
 - Hi ha un procés intern del sistema que periòdicament **monitoritza el nº de marcs lliures** (`frame_free`)
 - Si (`frame_free`) > REC, aleshores no cal fer res
 - Si (`frame_free`) < MIN aleshores:
 - S'escriuen en disc alguns processos complets (swap out) fins aplegar a REC marcs lliures
 - Es trien com víctimes aquells processos que duen més temps sense fer res (suspesos). Per ex., servidors que no hagen rebut cap petició.
 - Si $\text{MIN} \leq \text{frame_free} \leq \text{REC}$ aleshores:
 - Es cerquen processos amb un elevat nombre de marcs assignats i amb una baixa taxa de fallades de pàgina. S'aplica l'algorisme de reemplaçament per a "furtar-los" alguns marcs.
 - Se seleccionan diverses víctimes en cada procés. El nombre concret depén de cada sistema operatiu.

- **Gestió de la reserva en la hiperpaginació**
 - Si hiperpaginació, el nombre de marcs lliures baixa ràpidament
 - El procés monitor del nivell de la reserva ho detecta quan
 - Entre dues activacions successives d'aquest procés el nivell de `frame_free` baixa mes del que seria convenient.
 - Solucions:
 - Expulsar (swap out) processos complets fins aplegar a un nivell recomanable (REC).
 - Gestió utilitzada en OpenVMS i Windows.
 - Altres monitors alliberen un nombre constant de marcs en cada activació (si el nivell està per sota de REC).
 - Aleshores el monitor passa a activar-se amb major freqüència.
 - Gestió utilitzada en UNIX SVR4.

- **Contingut dels marcs de la reserva**

- Si el S.O. Tria una víctima el seu marc no s'utiliza d'immediat per a una altra pàgina el marc PASSA A LA RESERVA
 - Si les **pàgines víctimes tornen a ser referenciades** pel procés ràpidament
 - Hi ha una provabilitat alta de que encara es trobe disponibles en la reserva i es puguin reassignar sense necessitat de tornar-les a llegir del disc
 - El S.O. Recorda quin es el contingut de cada marc de la seua reserva
 - Si els marcs inclosos en la reserva corresponen a **pàgines modificades**
 - No son considerats per a atendre fallades de pàgina de forma immediata. Ja que han d'escriure el seu contingut en un fitxer o partició d'intercanvi
 - » Quan es supera cert mínim, totes eixes pàgines són escrites en el fitxer o partició d'intercanvi, d'una sola vegada
 - » Evitem fer “page out” per cada pàgina, minimitzan el nombre global d'escriptures necessàries.
 - Una vegada escrites les pàgines els seus marcs són lliures i poden utilitzar-se per a servir fallades de pàgina