

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 11 Memòria Virtual (I)

f SO

DISCA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

- **Objetius**

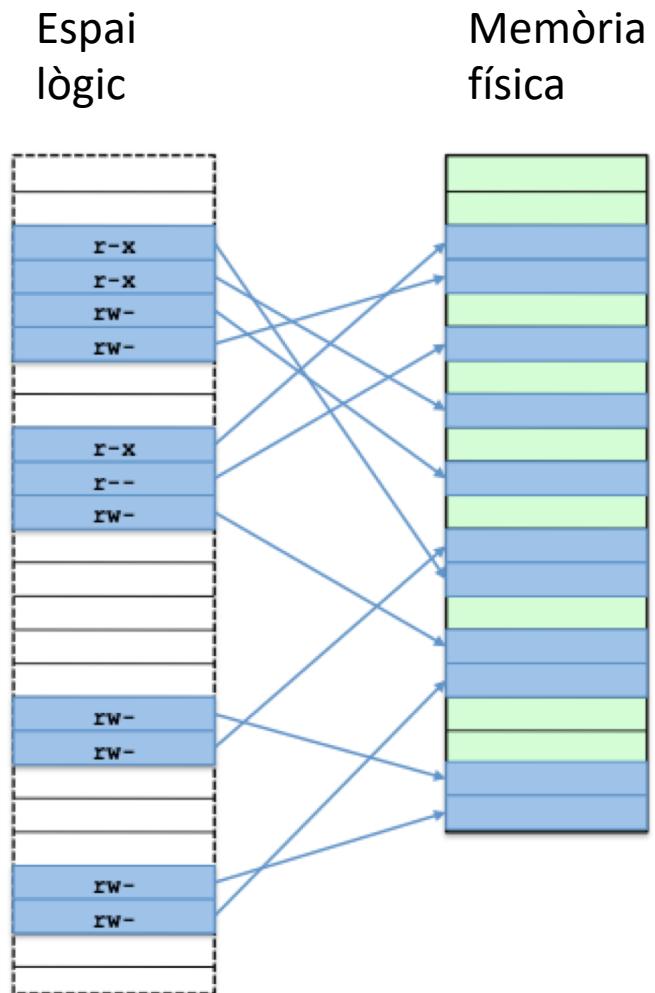
- Descriure els **avantatges** d'un sistema de Memòria Virtual i els seus efectes sobre el **rendiment** del sistema
- Entendre el concepte de **paginació sota demanda**
- Estudiar les **tècniques de reemplaçament de pàgina**

Bibliografia

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

- Contingut
 - **Objectiu de la Memòria Virtual**
 - Concepte de Memòria Virtual
 - Suport a la Memòria Virtual
 - Paginació per Demanda
 - Memòria Virtual i Gestió de Processos
 - Sèrie de Referències
 - Algorisme de Reemplaçament FIFO
 - Algorisme de Reemplaçament Òptim
 - Algorisme de Reemplaçament de LRU

- Gestió de memòria sense MV
 - El SO reserva per al procés **tota la memòria** prevista en el seu mapa, tot i que no la utilitze
 - Tots els accessos a la memòria principal duren el mateix temps
 - Quan el procés acaba, el SO reutilitza tots els marcs alliberats

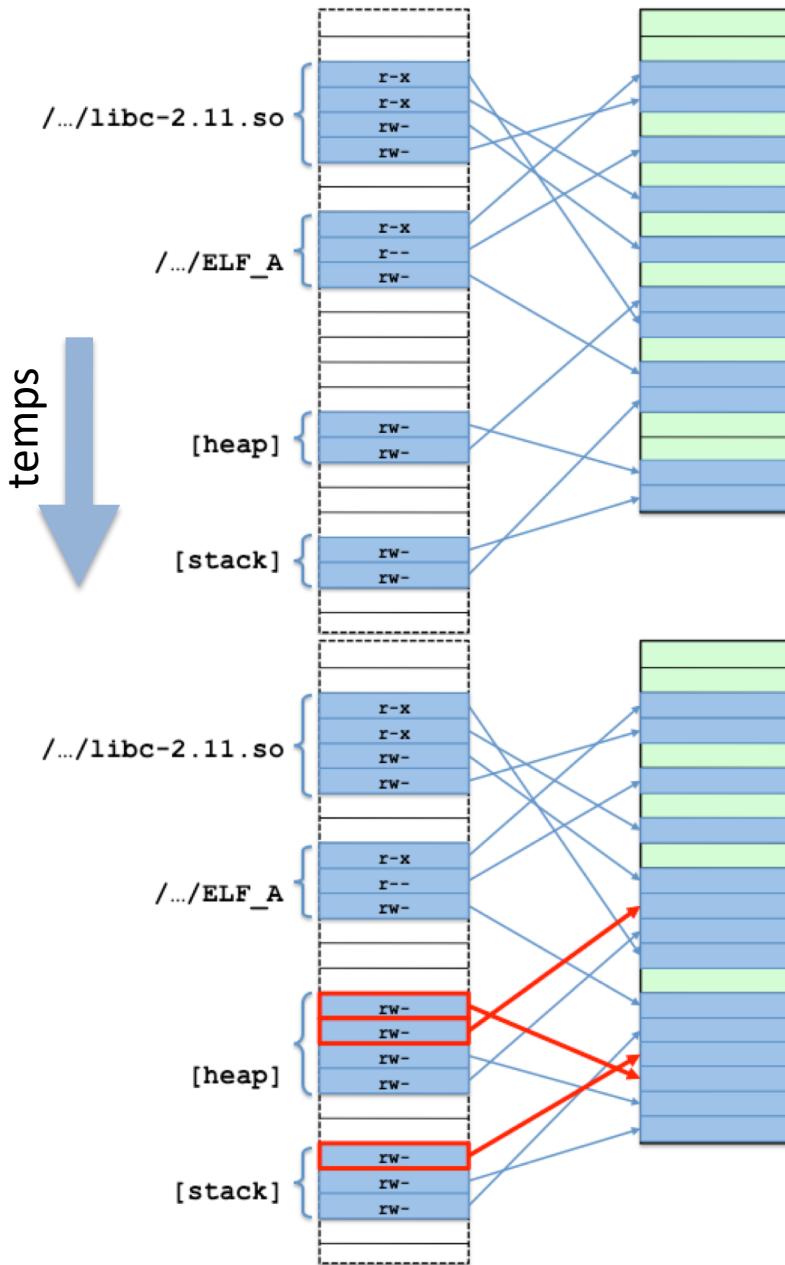


Marc Iliure,
disponible per a
altres processos

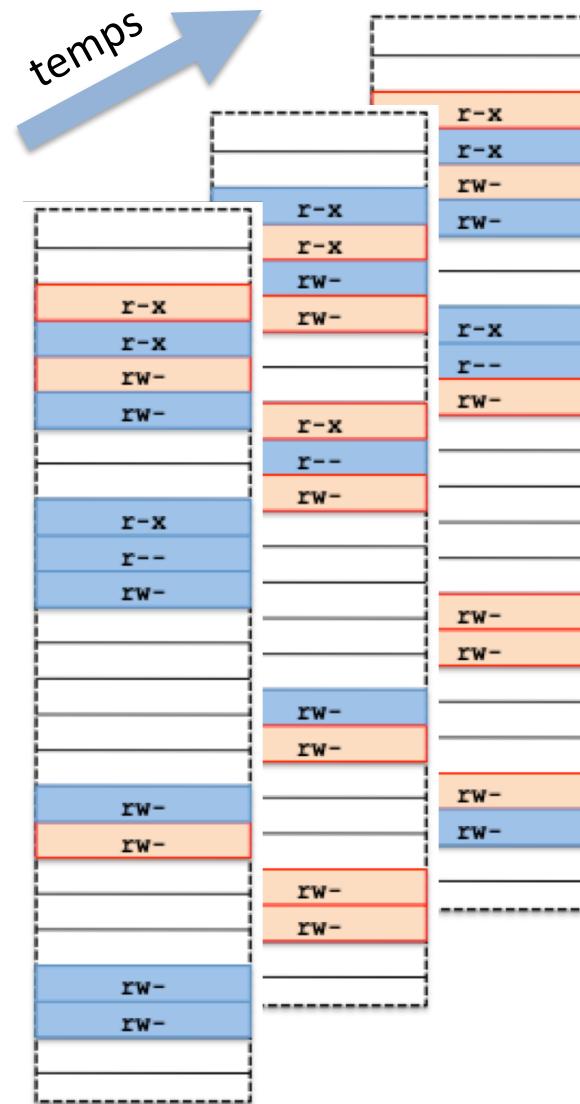
Objectiu de la Memòria Virtual

fso

- El mapa de memòria d'un procés evoluciona
 - Els processos tenen requeriments de memòria canviants
 - El compilador només pot preveure les instruccions i les variables globals del programa
 - El nombre de regions creixen i també la seu grandària
 - La pila creceix quan les funcions es criden unes a les altres
 - Les regions de *heap* apareixen i creixen per efecte de les crides de reserva de memòria dinàmica
 - La creació de nous fils exigeix memòria per a les seues variables locals

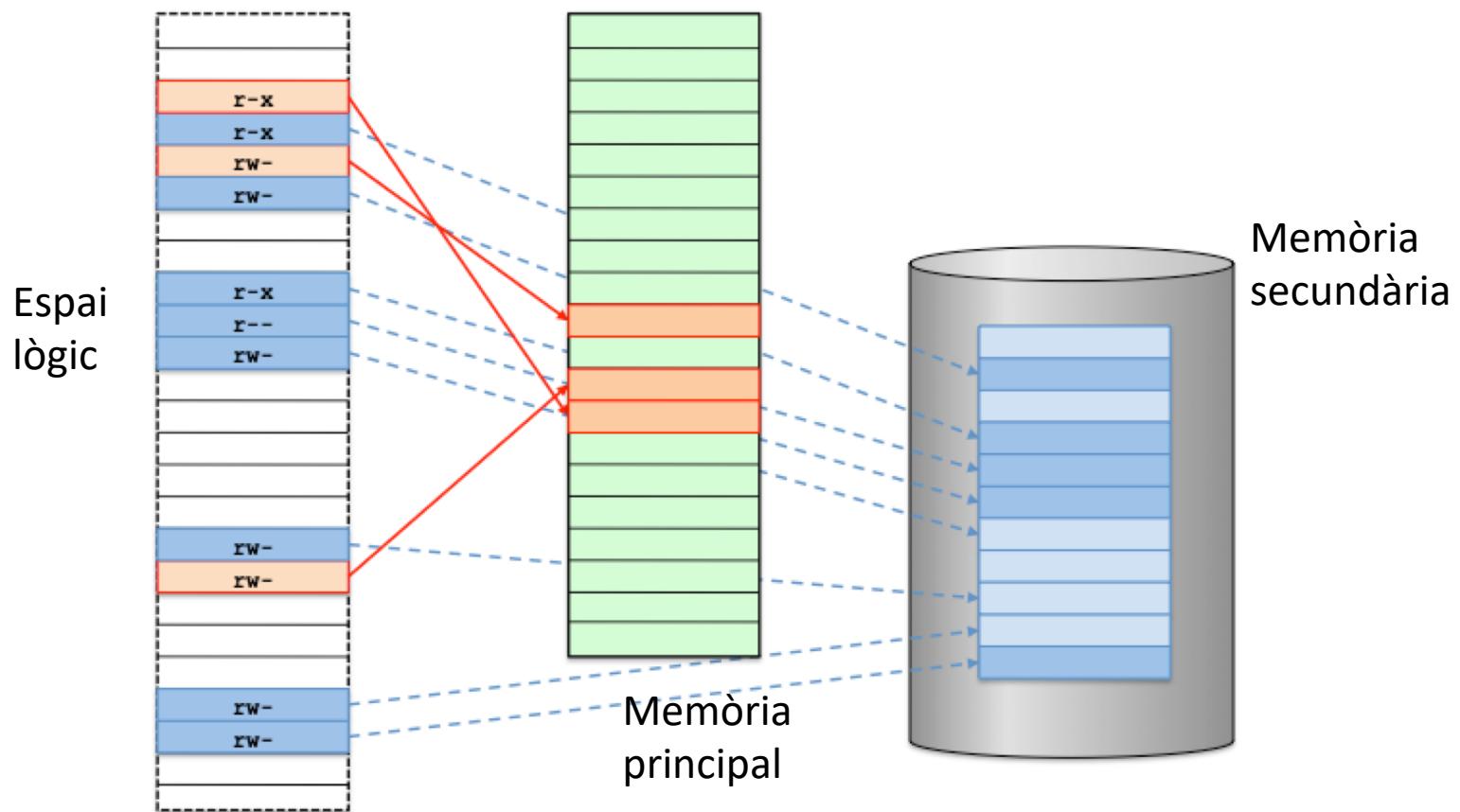


- Principi de localitat de referència
 - Un procés pot ocupar molta memòria lògica, però els accessos tendeixen a concentrar-se en zones *calentes*
 - En un interval de temps donat, es llegeixen les instruccions d'uns pocs mètodes/funcions, es llegeixen i s'escriuen reiteradament unes poques variables
 - A mesura que el procés evoluciona, els punts calents canviaran de lloc (altres funcions, altres variables)



- **Gestió de memòria amb MV**

- El SO gestiona l'assignació de memòria del procés, per a que només les zones calentes es troben a la memòria física
- La resta del espai lògic del procés es troba en la memòria secundària (habitualment el disc): àrea de *swap*



- Contingut
 - Objectiu de la Memòria Virtual
 - **Concepte de Memòria Virtual**
 - Suport a la Memòria Virtual
 - Paginació per Demanda
 - Memòria Virtual i Gestió de Processos
 - Sèrie de Referències
 - Algorisme de Reemplaçament FIFO
 - Algorisme de Reemplaçament Òptim
 - Algorisme de Reemplaçament de LRU

- **La tecnologia base de la MV**

- La MV combina memòria principal amb memòria secundària (habitualment disc)
- La memòria principal està formada per paraules (32, 64 bits, etc) adreçables pel processador, i el seu temps d'accés es mesura en ns
 - Cada cicle d'instrucció suposa un o més accessos a la memòria principal
 - Cada accés (lectura de una instrucció, lectura o escriptura d'una dada) transfereix una paraula entre la memòria i el processador
- La memòria secundària està organitzada en blocs (512, 4096 bytes, etc) accessibles a través d'un adaptador i el seu temps d'accés es mesura en ms ($1 \text{ ms} = 10^6 \text{ ns}$)
 - Para una operació amb la memòria secundària, el processador ha d'executar moltes instruccions
 - La transferència d'una pàgina entre la memòria principal i la secundària es realitza en una única operació d'E/S

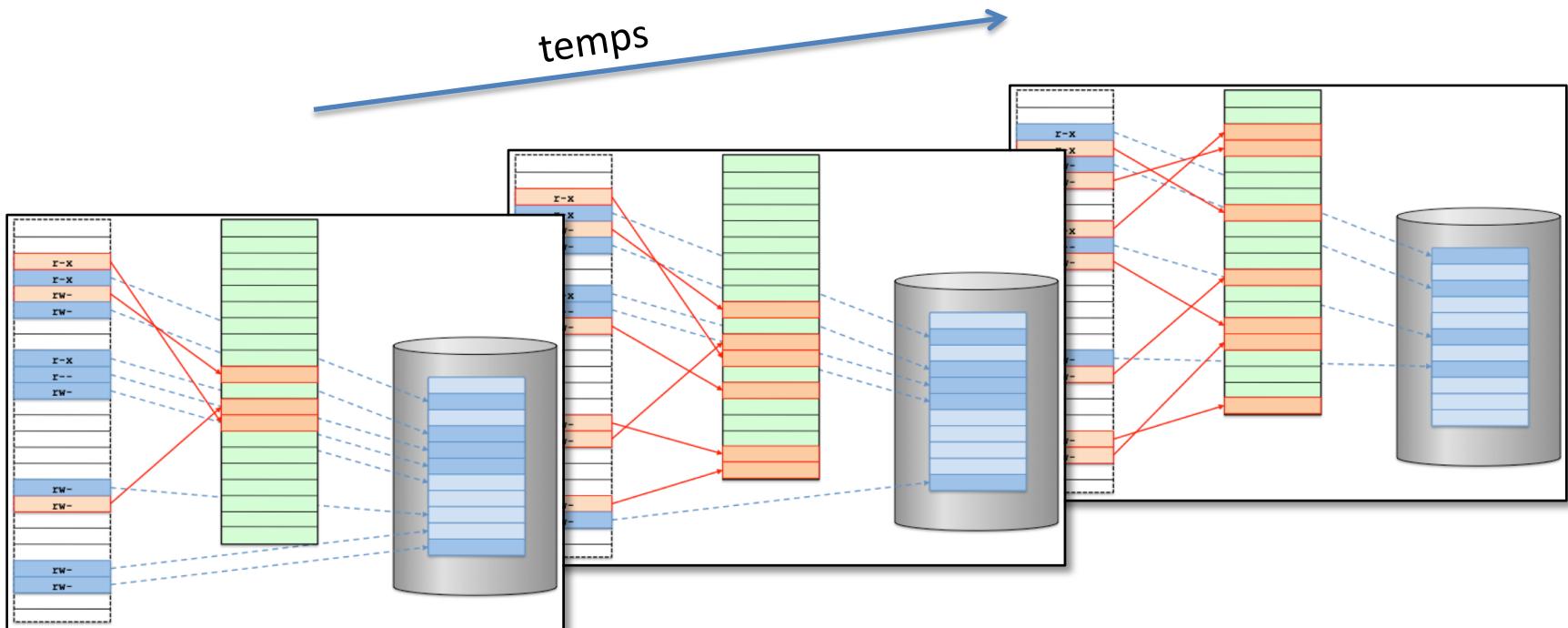
- **Esquema de la MV**

- El SO gestiona una assignació dispersa de memòria (habitualment paginació)
- El SO gestiona el àrea de swap.
 - Segons el cas, en una partició dedicada en el disc dur o un arxiu en el sistema d'arxius comú
- Cadascuna de les pàgines del espai lògic de un procés pot trobar-se en dos estats
 - Vàlida: la pàgina se troba en un marc
 - Invàlida: la pàgina està en el àrea de swap
- Dues situacions durant cada accés a la memòria principal:
 - Encert(situació més freqüent): referència a pàgina vàlida.
 - Fallada: referència a pàgina invàlida. Es necessari la taula d'assignació i, segons el cas, transferir una o més pàgines entre la memòria física i el disc.

Concepte de Memòria Virtual

fso

- Paganació per demanda = **Paganació + Intercanvis** (swapping) entre memòria principal i secundària
 - Les **pàgines** es carreguen **en memòria** quan es fa referència a elles durant l'execució del procés
 - Cada **pàgina** carregada en memòria té assignat un **marc**
 - Quan un procés deixa d'accédir a una pàgina, el SO acabarà transferint-la a la memòria secundària per a alliberar espai
 - Les pàgines que mai s'accedeixen sinó es carreguen

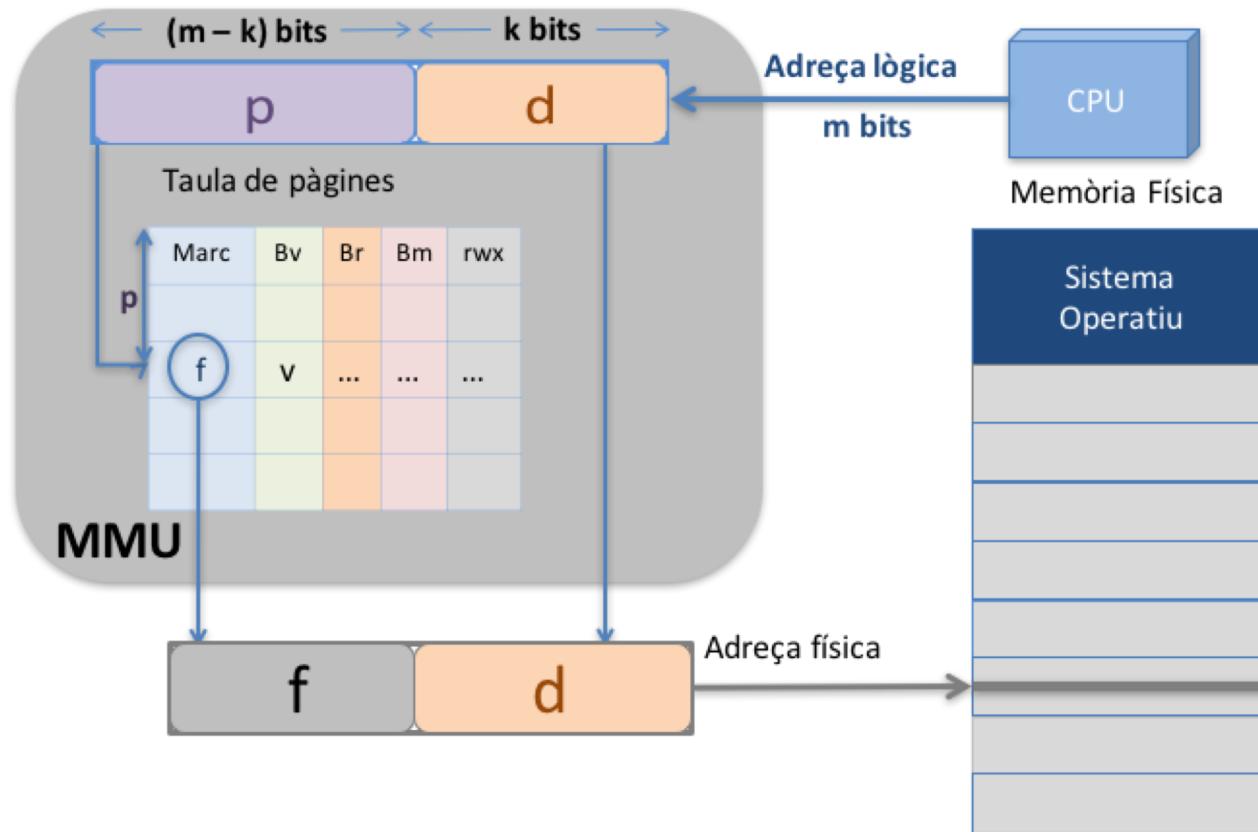


- Beneficis de la paginació (amb o sense MV)
 - Es poden restringir els accessos a cada pàgina
 - Es detecten errors de programa i accessos maliciósos
 - Es poden compartir pàgines entre processos:
 - Por exemple, les instruccions de les biblioteques comuns
 - Permet compartir variables entre processos
- Beneficis de la Memòria Virtual
 - Estalvia memòria:
 - **Augmenta** el grau de **multiprogramació**
 - Permet processos de **major grandària**
 - Facilita la gestió de la memòria dinàmica
- Cost de la Memòria Virtual
 - Creix el temps de retorn dels processos perquè el tractament d'algunes fallades de pàgina els suspèn
 - L'àrea de swap ocupa espai en disc
 - Augmenta la càrrega de treball de la memòria secundària
 - Major **complexitat** en el disseny del sistema operatiu
 - El PCB ha de protegir la taula d'assignació de cada procés

- Contingut
 - Objectiu de la Memòria Virtual
 - Concepte de Memòria Virtual
 - **Suport a la Memòria Virtual**
 - Paginació per Demanda
 - Memòria Virtual i Gestió de Processos
 - Sèrie de Referències
 - Algorisme de Reemplaçament FIFO
 - Algorisme de Reemplaçament Òptim
 - Algorisme de Reemplaçament de LRU

Suport a la MV

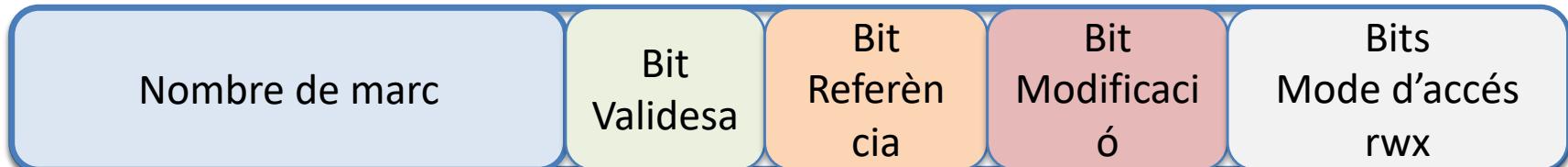
- L'organització de la taula de pàgines no canvia amb la MV
 - Cada entrada de la taula consisteix en un descriptor de pàgina
 - La MMU utilitza el nombre de pàgina per a indexar la taula



- Els descriptores han de tenir més informació:

- Descriptor de pàgina per a la MV:
 - Cada descriptor de pàgina té
 - **Nombr de marc** on està ubicada la pàgina en memòria (si és vàlida)
 - **Bit de Validesa**: Indica si una pàgina està mapejada en memòria. Suporta la paginació per demanda.
 - **Bit de Referència**: Indica si la pàgina ha sigut accedida. Necessari per a algorisme de segona oportunitat.
 - **Bit de Modificació**: Indica si la pàgina ha sigut accedida per a escriptura.
 - **Bits de Modo de Accés**: només lectura, lectura-escriptura, execució, ..

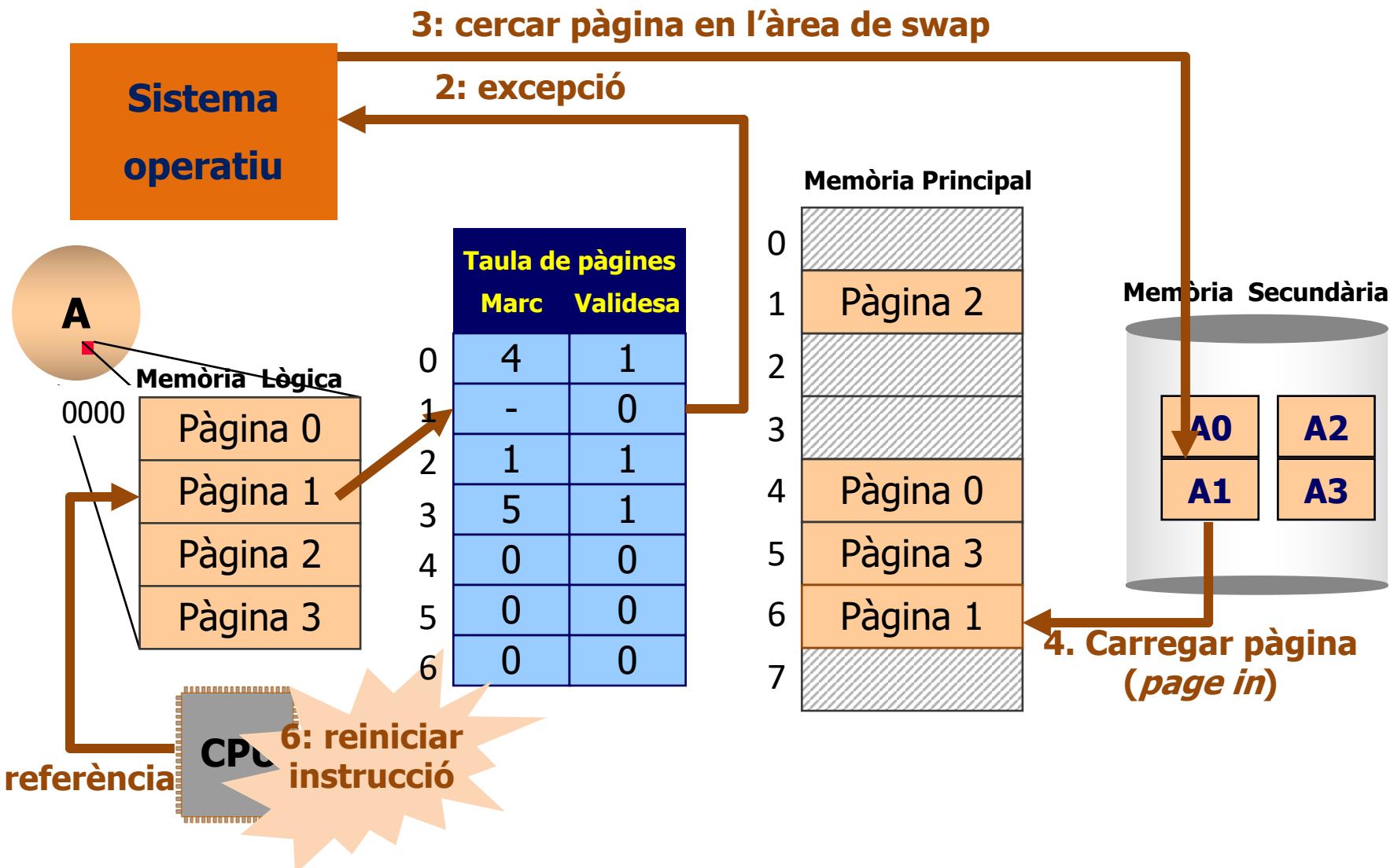
Descriptor de pàgina



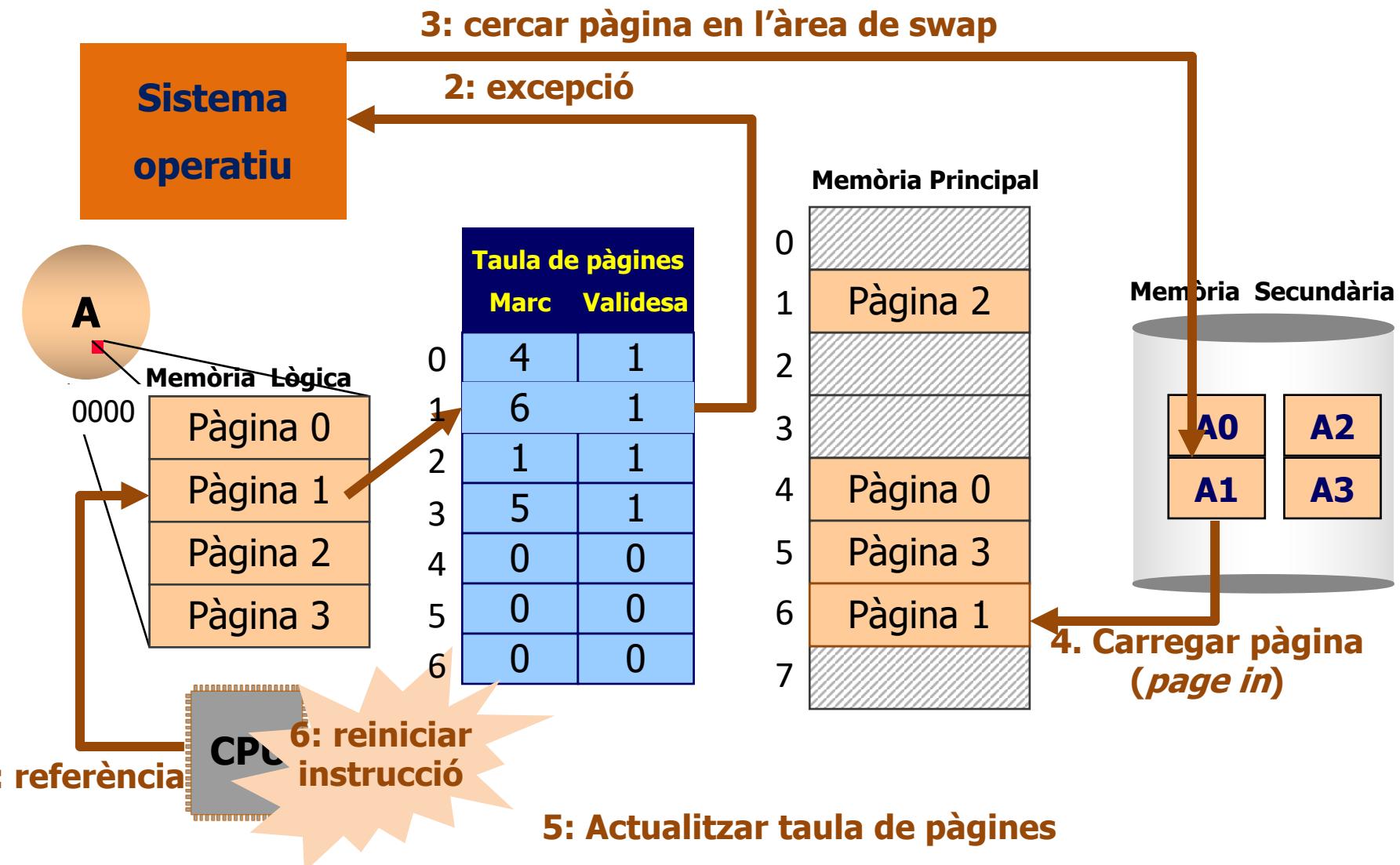
- **Les excepcions de la MMU**
 - Accés sense incidents (el més habitual)
 - Accés correcte a pàgina vàlida: el descriptor subministra l'adreça física
 - El procés en execució no canvia d'estat
 - La MMU provocarà excepció de **fallada de pàgina** (i el SO intervindrà) en, almenys, dos casos:
 - Accés a pàgina invàlida. El SO tracta l'excepció com a senyal de demanda
 - Les transferències entre disc i memòria per a resoldre fallades de pàgina suspenen el procés
 - **Violació d'accés (*segmentation fault*)**: accés a una pàgina fora del mapa o accés no permès a una pàgina
 - Normalment, el SO acaba el procés

- Contingut
 - Objectiu de la Memòria Virtual
 - Concepte de Memòria Virtual
 - Suport a la Memòria Virtual
 - **Paginació per Demanda**
 - Memòria Virtual i Gestió de Processos
 - Sèrie de Referències
 - Algorisme de Reemplaçament FIFO
 - Algorisme de Reemplaçament Òptim
 - Algorisme de Reemplaçament de LRU

- Fallada de pàgina: cas de Pàgina en Disc



- Fallada de pàgina: cas de Pàgina en Disc

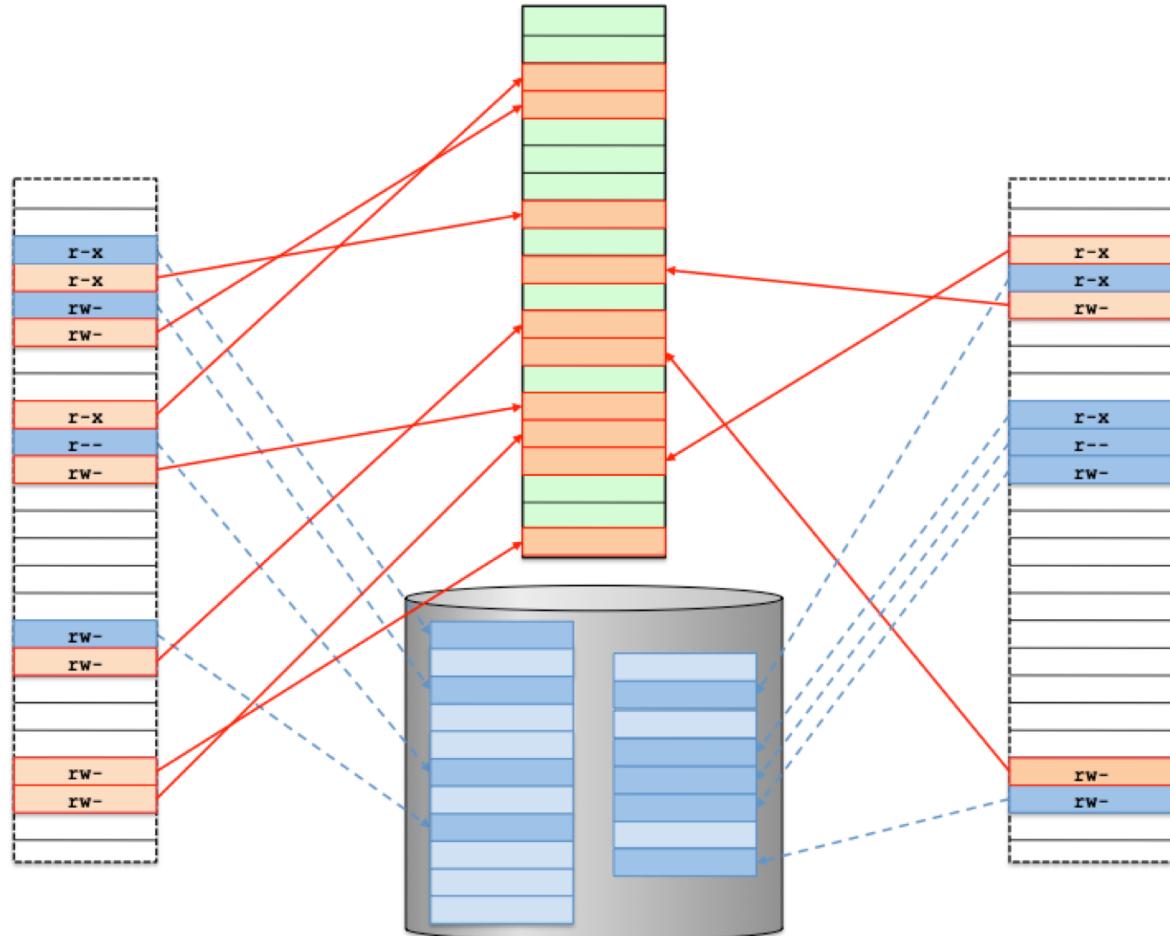


- Algorisme de fallada de pàgina: cas de pàgina en disc
 - Trobar la **pàgina demandada** en disc
 - Trobar un **marc lliure**:
 - Si existeix un marc lliure, utilitzar-lo.
 - Si no existeix marc lliure,
 - Executar algorisme de reemplaçament de pàgines i seleccionar víctima
 - Si la víctima té el bit de modificació a 1, escriure la víctima en disc (page out).
 - En la víctima: Actualitzar taula de pàgines, amb bit de validesa= 0
 - Actualitzar la taula de marcs.
 - **Llegir la pàgina demandada en disc** (page in) i ubicar-la en el marc lliure
 - **Actualitzar taula de pàgines** del procés que genera el fallada
 - Actualitzar la taula de marcs lliures
 - El procés d'usuari passa a preparat
 - quan obtinga de nou la CPU, continuarà la seu execució en la instrucció que va provocar el fallada de pàgina.

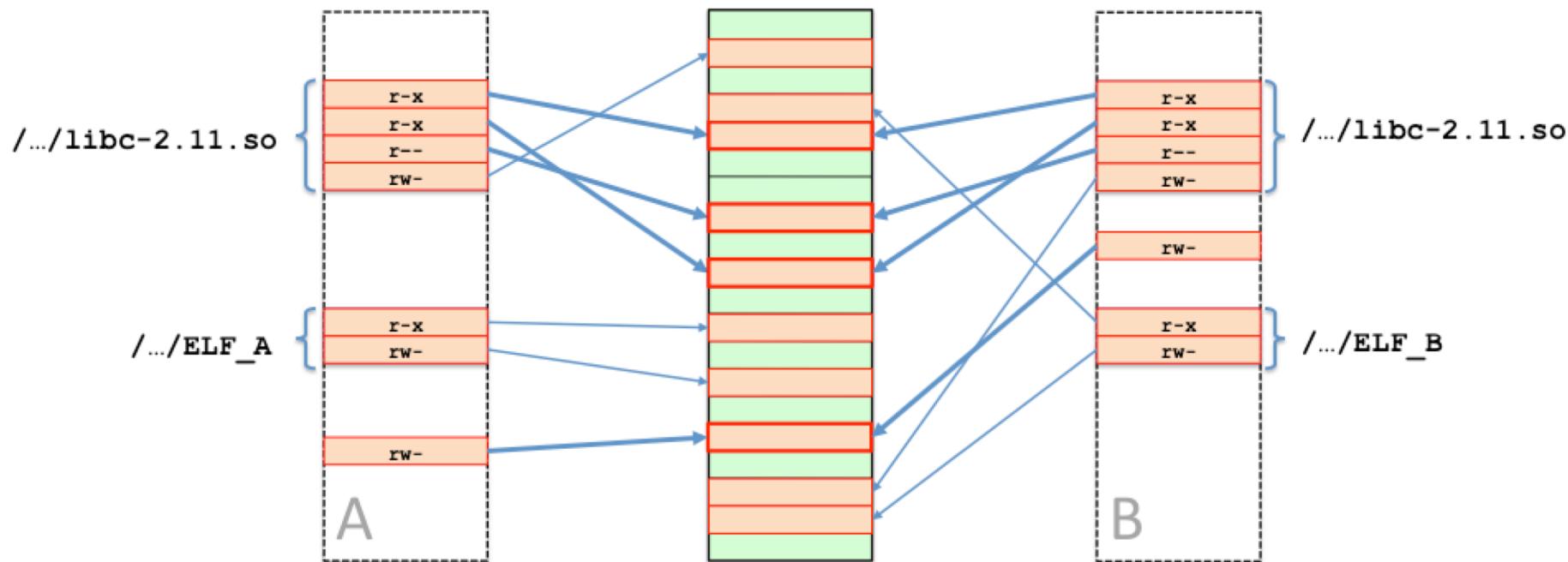
- **Reemplaçament de Pàgina**
 - Són necessaris quan la memòria principal està completament ocupada i es produceix un fallada de pàgina:
 - Una pàgina ubicada en memòria principal, denominada **víctima**, ha de deixar el seu **marc** a la pàgina demandada
 - Si pàgina víctima té el bit de modificació a 1 cal que **salvar la víctima a disc (page out)**
 - S'ubica la **pàgina demandada sobre el marc** de la víctima **(page in)**
 - Existeixen diversos algorismes per a seleccionar la víctima

- Contingut
 - Objectiu de la Memòria Virtual
 - Concepte de Memòria Virtual
 - Suport a la Memòria Virtual
 - Paginació per Demanda
 - **Memòria Virtual i Gestió de Processos**
 - Sèrie de Referències
 - Algorisme de Reemplaçament FIFO
 - Algorisme de Reemplaçament Òptim
 - Algorisme de Reemplaçament de LRU

- La MV afavoreix la concorrència
 - Al mantenir en la memòria principal només les zones calentes dels processos, aprofitarà millor la seu capacitat
 - Exemple: el procés A ocupa 11 pàgines i B ocupa 8. I només ocupen 10 marcs!



- Compartició de marcs
 - El SO pot ubicar pàgines de distints processos en el mateix marc
 - Útil per a codi executable de biblioteques comuns i per a que els processos puguen compartir variables i arxius projectats
- Exemple: dos processos A i B (creats a partir dels executables ELF_A i ELF_B) comparteixen quatre marcs
 - Tres pàgines de la biblioteca libc-2.11.so
 - Una pàgina sense suport amb variables compartides



- UNIX: les crides fork i exec
 - Una crida a *exec* invalida totes les pàgines de la taula del procés
 - Les fallades en els accessos posteriors actualitzaran els descriptors amb el mapa de memòria del nou executable
 - Una crida a *fork* clona la taula del pare en el procés recentment creat
 - Els processos pare i fill disposen de les pàgines de només lectura (codi executable i constants) en marcs compartits
 - El SO ubicarà les pàgines amb permís d'escriptura (variables, pila, heap) en marcs distints. L'assignació és progressiva (*copy-on-write*): quan un procés fa referència per primera vegada a una d'elles, el SO busca un marc nou.

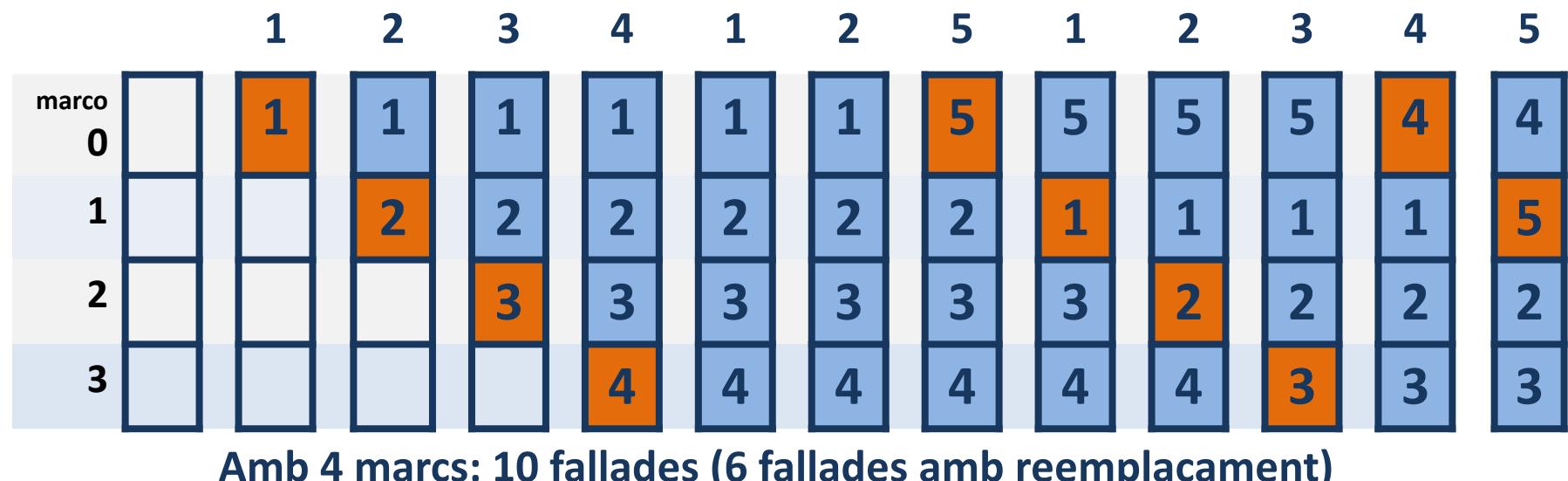
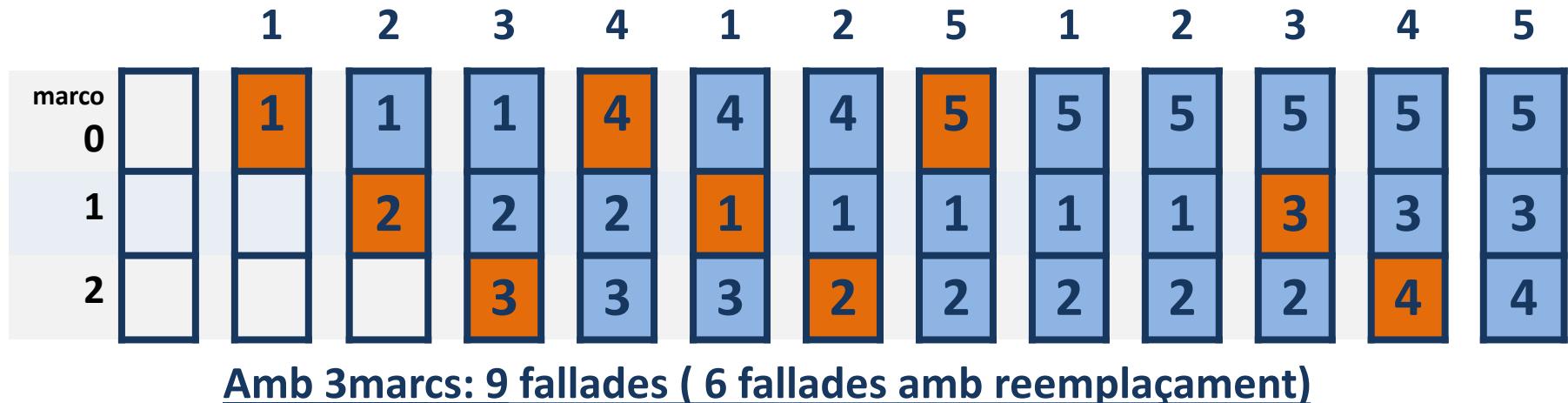
- Contingut
 - Objectiu de la Memòria Virtual
 - Concepte de Memòria Virtual
 - Suport a la Memòria Virtual
 - Paginació per Demanda
 - Memòria Virtual i Gestió de Processos
 - **Sèrie de Referències**
 - Algorisme de Reemplaçament FIFO
 - Algorisme de Reemplaçament Òptim
 - Algorisme de Reemplaçament de LRU

- **Sèrie de Referència:**
 - Seqüència de pàgines que han sigut **accedides** durant cert període de temps
 - De cada adreça lògica emesa pel processador s'obté el seu nombre de pàgina
 - S'eliminen les repetitions: Múltiples referències consecutives a una mateixa pàgina es representa per una sola referència
- **Ejemplo:**
 - Sistema amb pàgines de 1000 paraules:
 - Adreces lògiques del procés P referenciades
2500, 5100, 5234, 1800, 1432, 4388, 2124, 8216, 8498
 - Pàgines referenciades
2, 5, 5, 1, 1, 4 , 2 , 8, 8
 - La sèrie de referències d'aquesta seqüència és:
2, 5, 1, 4, 2, 8

- Contingut
 - Objectiu de la Memòria Virtual
 - Concepte de Memòria Virtual
 - Suport a la Memòria Virtual
 - Paginació per Demanda
 - Memòria Virtual i Gestió de Processos
 - Sèrie de Referències
 - **Algorisme de Reemplaçament FIFO**
 - Algorisme de Reemplaçament Òptim
 - Algorisme de Reemplaçament de LRU

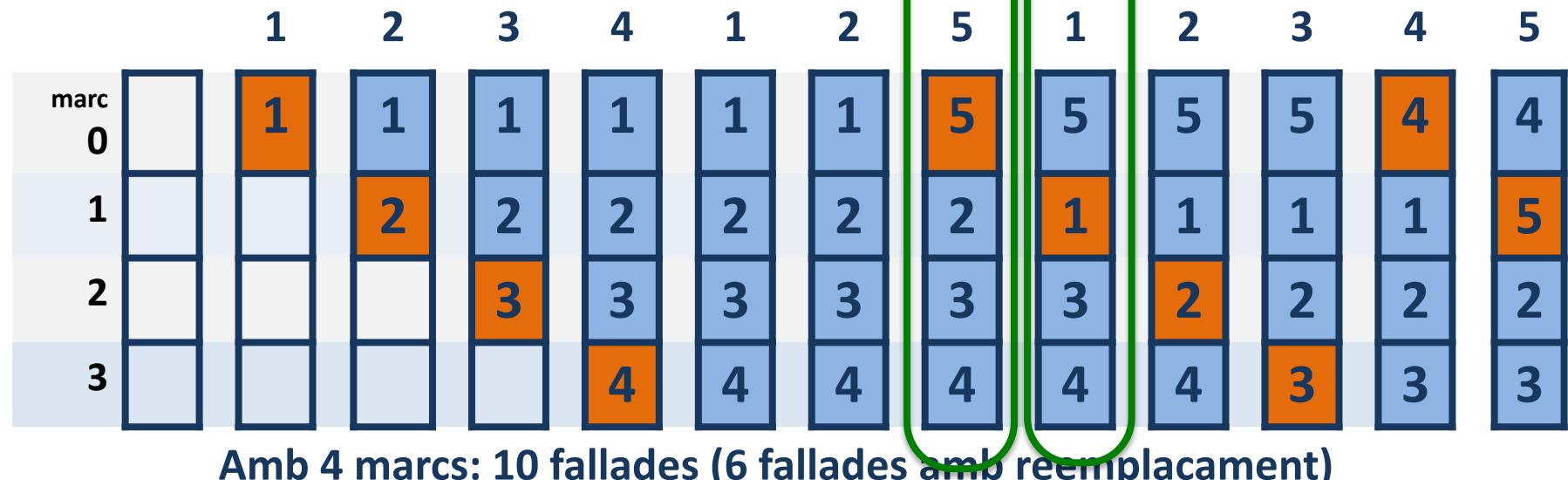
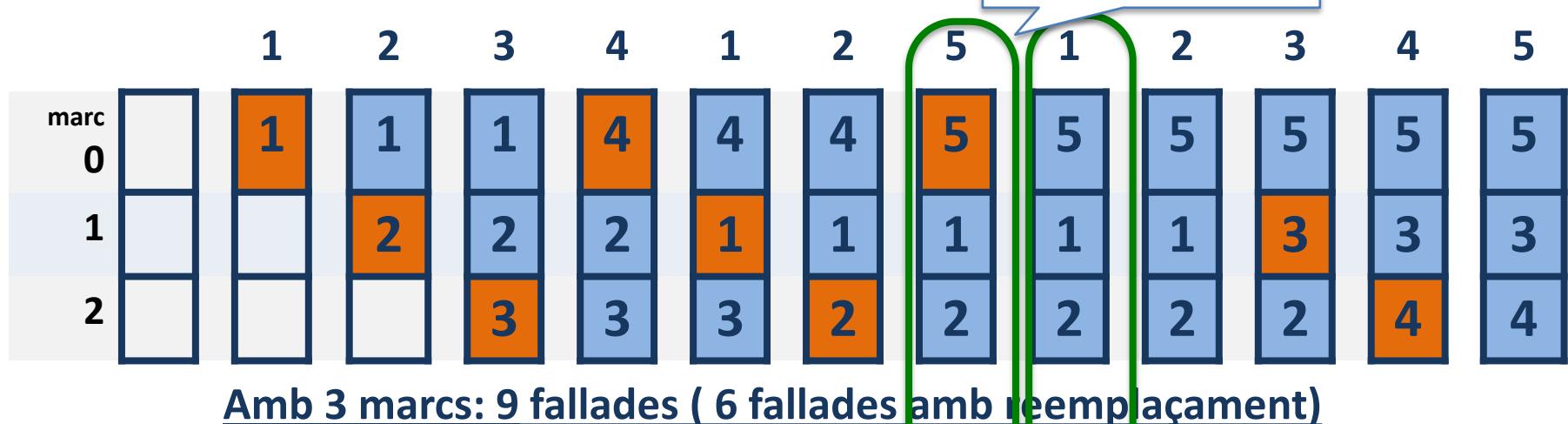
Algorisme FIFO

- Pàgina víctima: la que lleva més temps cargada en memòria.
- Presenta la anomalia de Belady



Algorisme FIFO

- Pàgina víctima: la que duu més temps carregada en memòria.
- Presenta l'anomalia de Belady

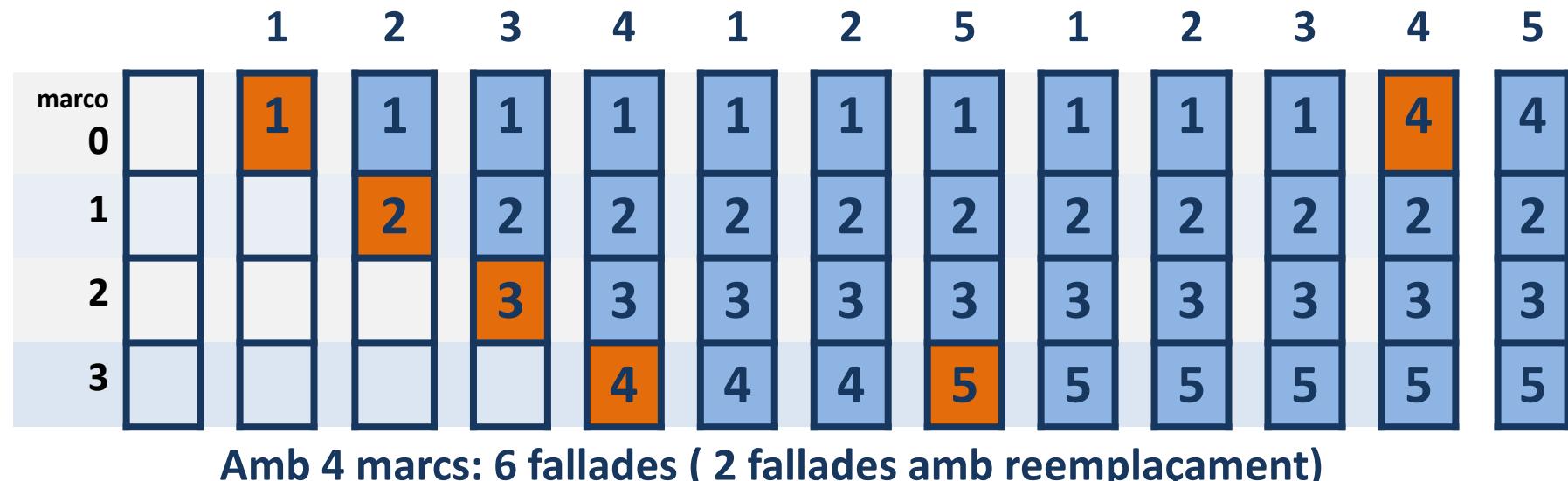
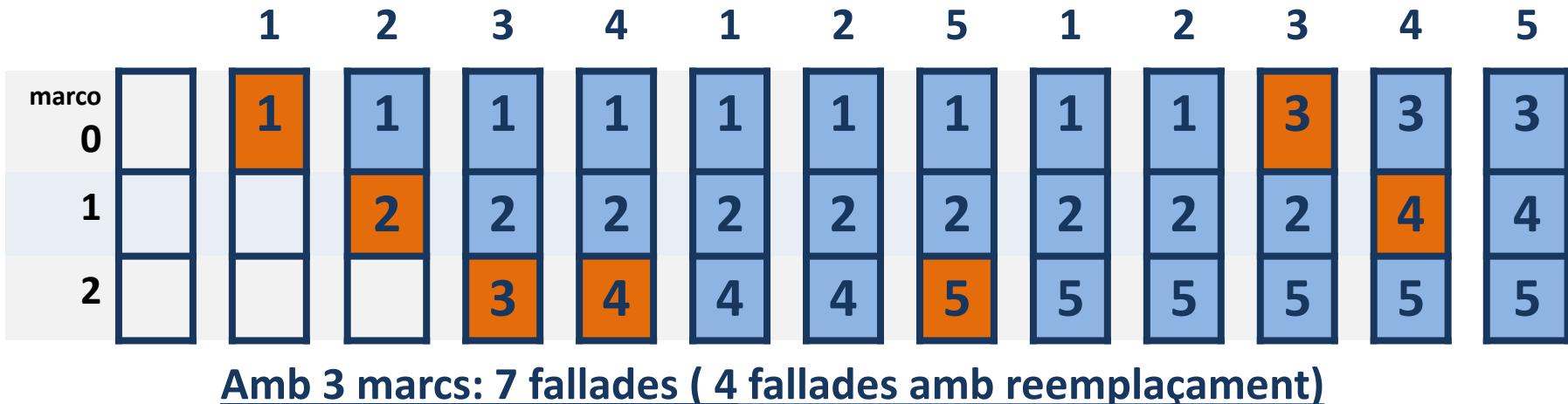


- **Algorismes de Pila**
 - Garanteixen que un conjunt de **pàgines** mantingut **amb N marcs** és un **subconjunt** del que es manté **amb N+1 marcs**
- Els algorismes de **Pila NO** presenten l'anomalia de **Belady**

- Contingut
 - Objectiu de la Memòria Virtual
 - Concepte de Memòria Virtual
 - Suport a la Memòria Virtual
 - Paginació per Demanda
 - Memòria Virtual i Gestió de Processos
 - Sèrie de Referències
 - Algorisme de Reemplaçament FIFO
 - **Algorisme de Reemplaçament Òptim**
 - Algorisme de Reemplaçament de LRU

Algorisme Òptim

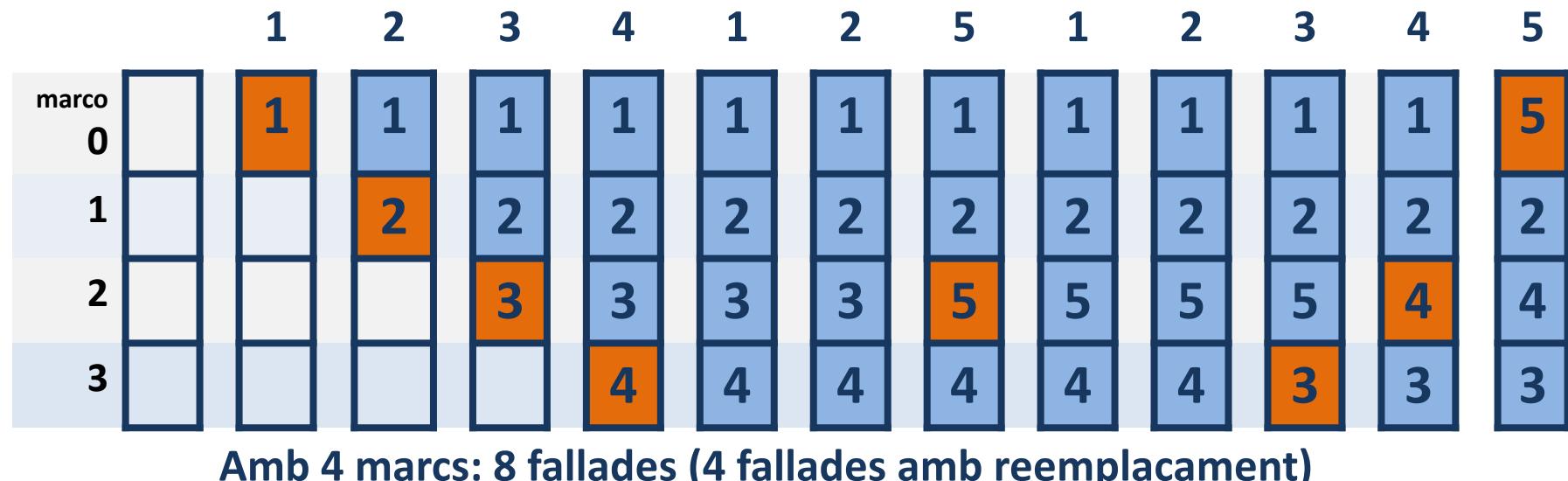
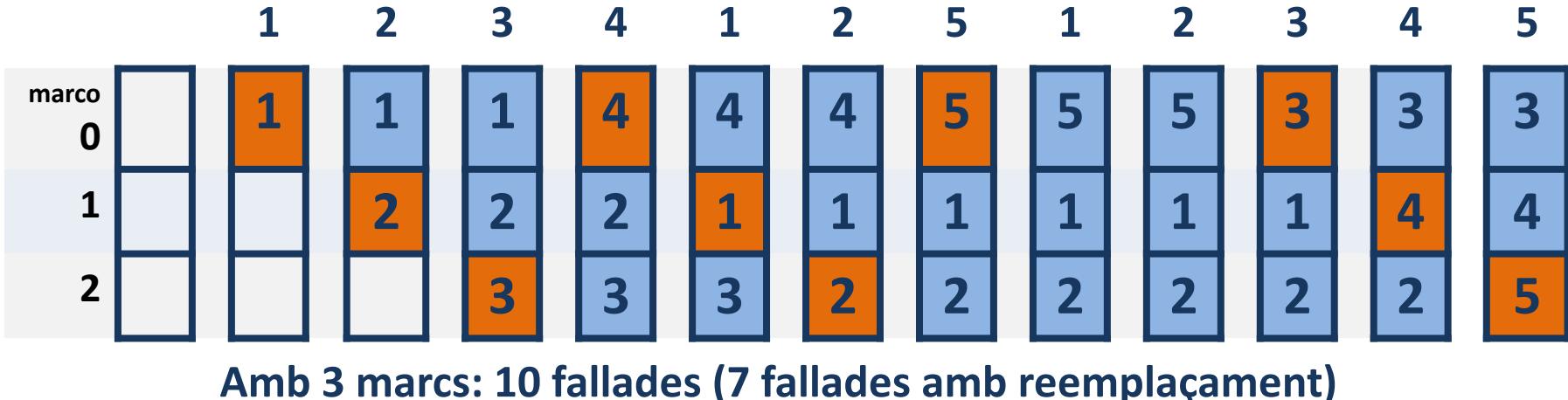
- Pàgina **víctima**: aquella que **més temps tardarà** en ser **referenciada**
- Nombre de fallades mínim. Implementació impossible



- Contingut
 - Objectiu de la Memòria Virtual
 - Concepte de Memòria Virtual
 - Suport a la Memòria Virtual
 - Paginació per Demanda
 - Memòria Virtual i Gestió de Processos
 - Sèrie de Referències
 - Algorisme de Reemplaçament FIFO
 - Algorisme de Reemplaçament Òptim
 - Algorisme de Reemplaçament de LRU**

Algorisme LRU

- Pàgina víctima: Aquella que fa més temps que ha sigut referenciada
- És un algorisme de Pila

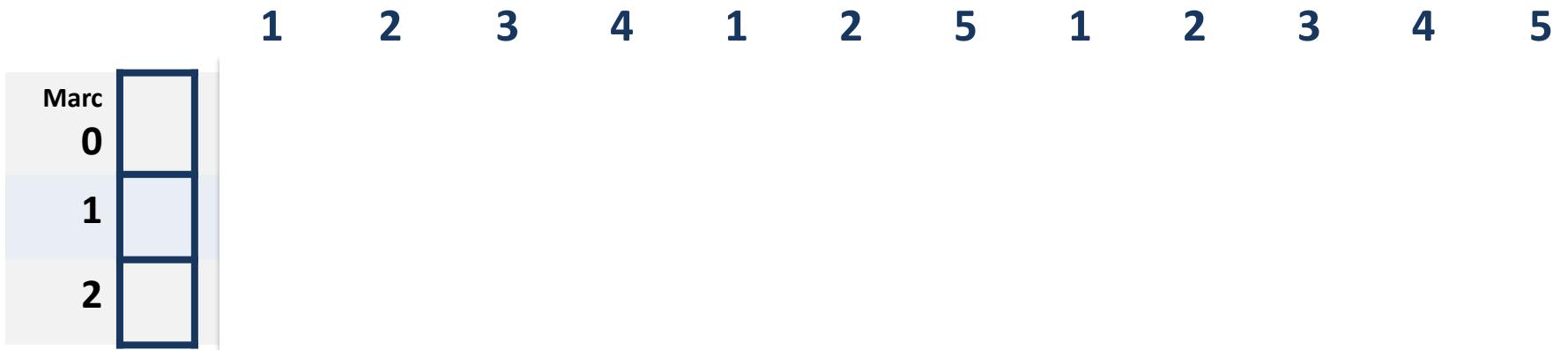


- Implementacions del LRU
 - Mitjançant comptadors:
 - Cada pàgina té un comptador associat
 - Mitjançant llista ordenada de pàgines referenciades
 - Quan se referència una pàgina es fica al final
 - La pàgina víctima es la primera de la llista
- Anàlisis d'eficiència
 - Avantatges:
 - Bona aproximació a l'òptim
 - Inconvenients:
 - Difícil de implementar
 - Solució:
 - Algorismes d'aproximació al LRU

- Seqüència del FIFO

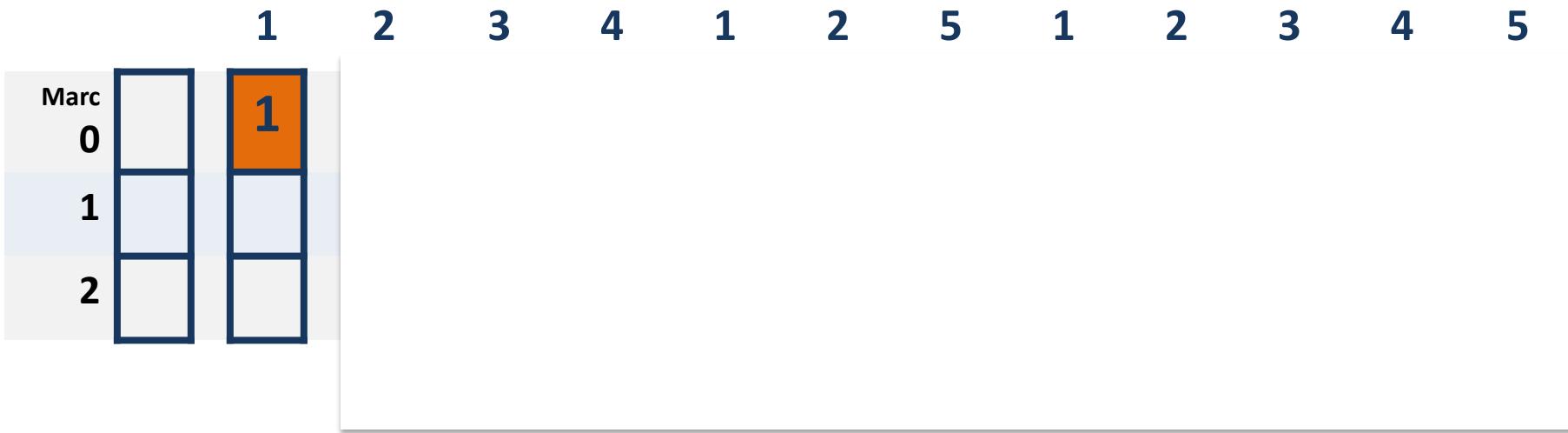
Algorisme FIFO

- Pàgina víctima: Aquella que **duu més temps carregada en memòria**.
- Presenta l'**anomalia de Belady**



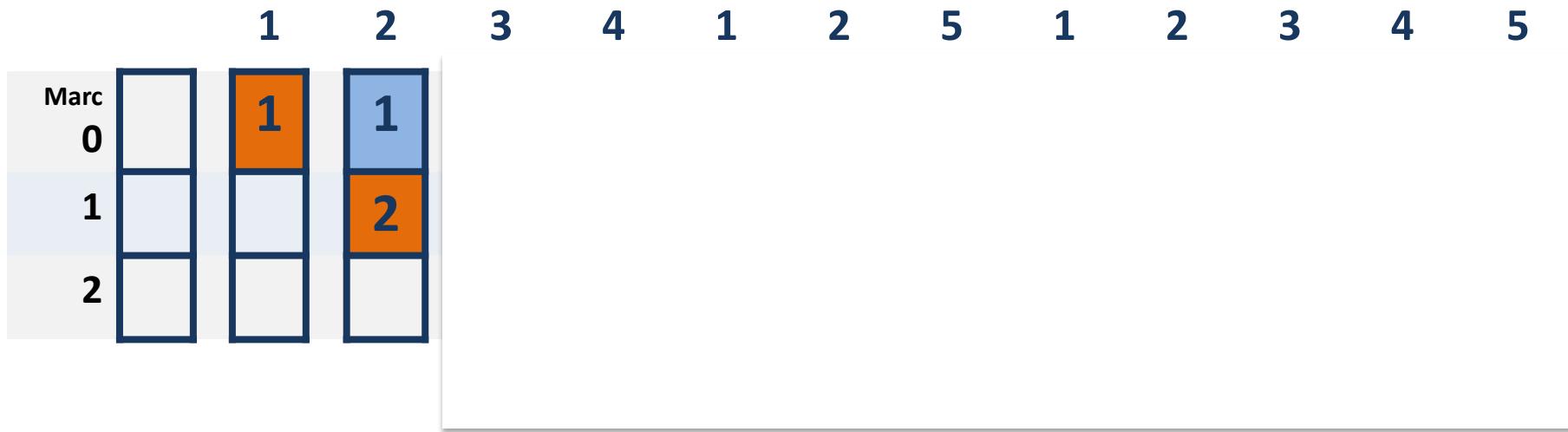
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'**anomalia de Belady**



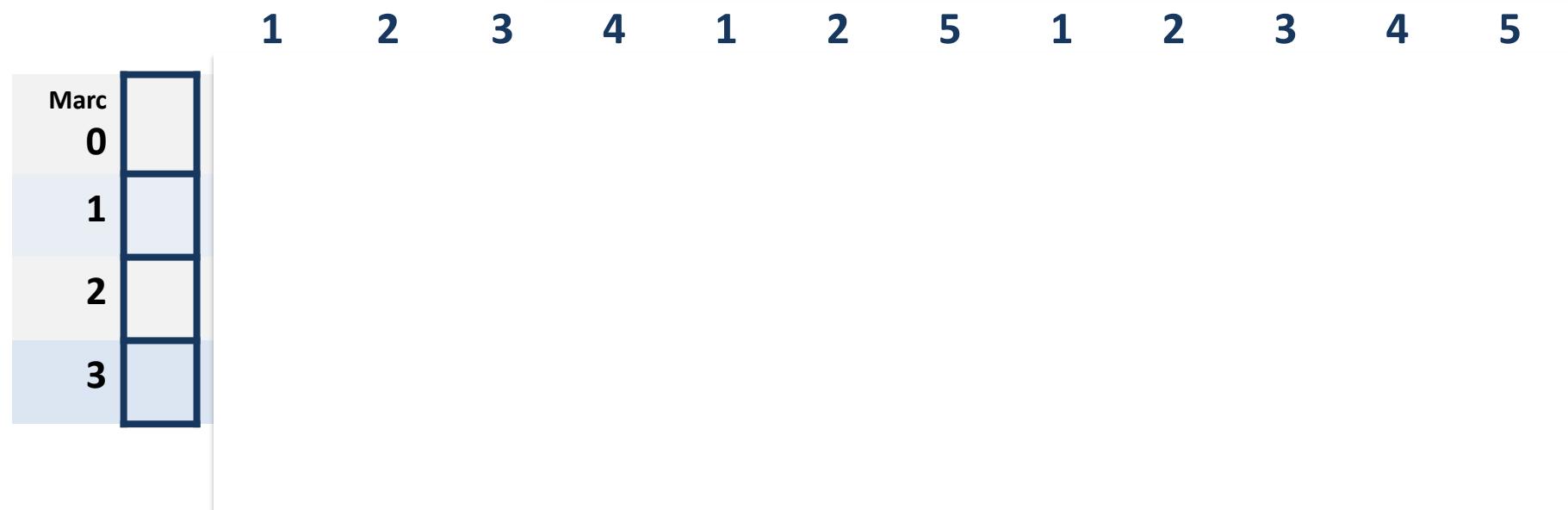
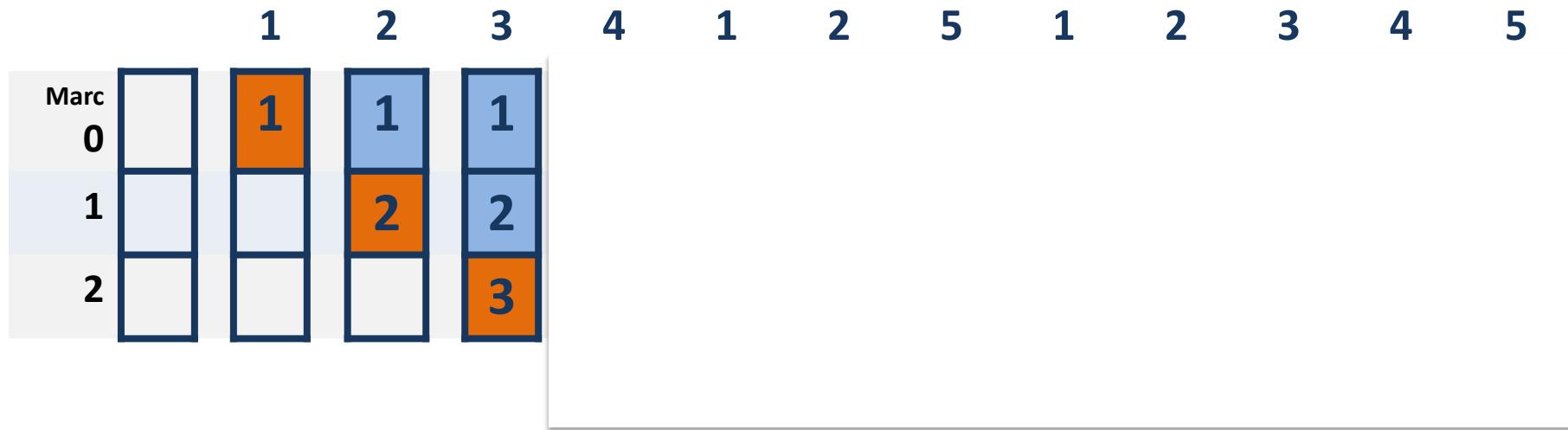
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'**anomalia de Belady**



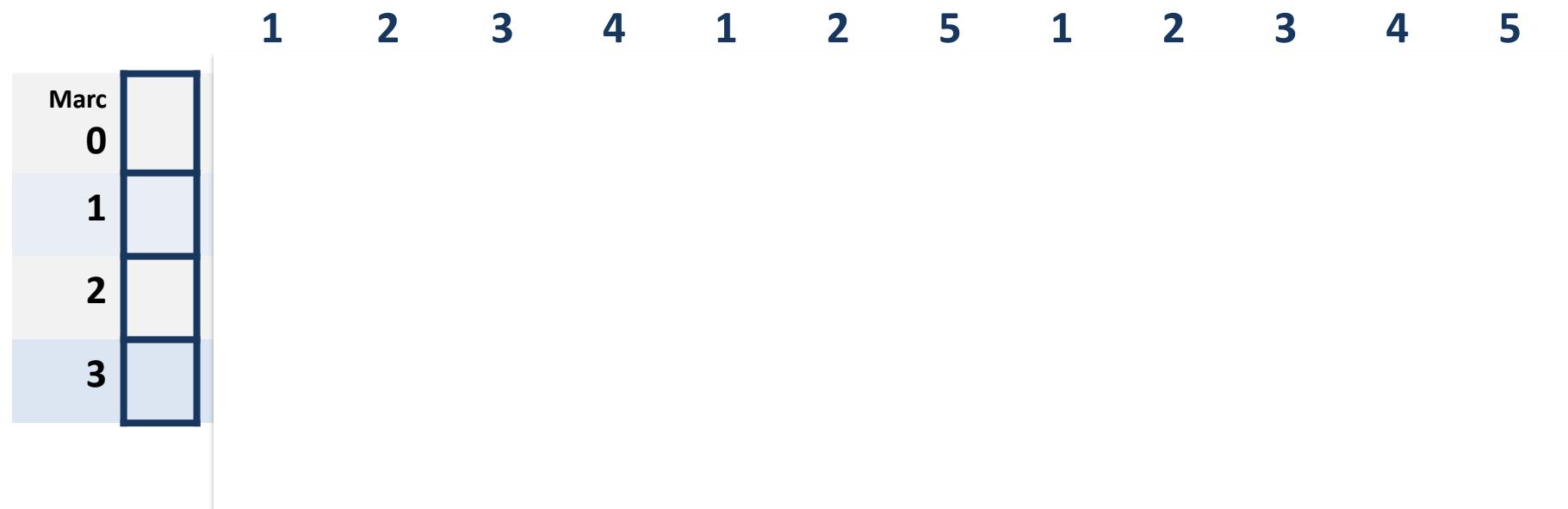
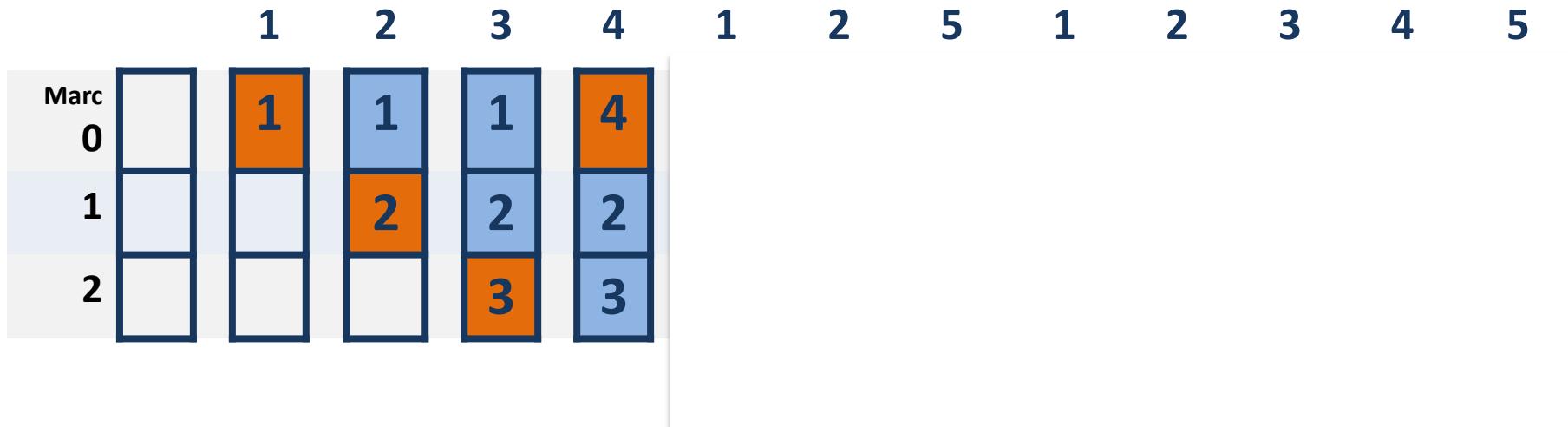
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'**anomalia de Belady**



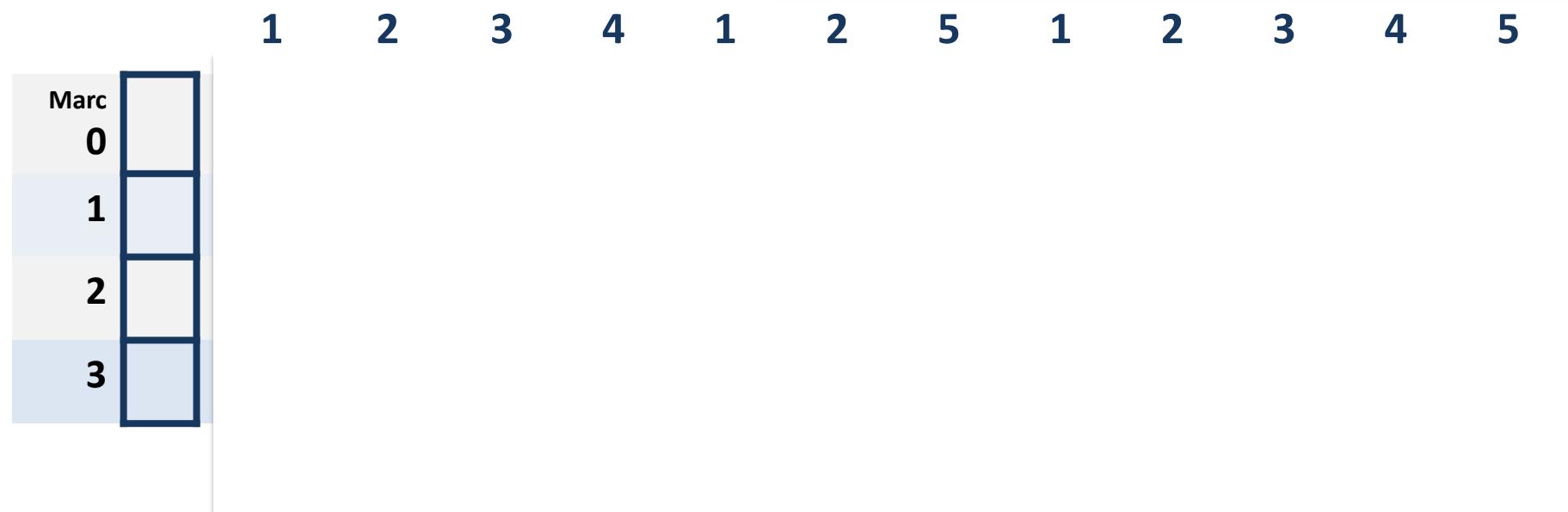
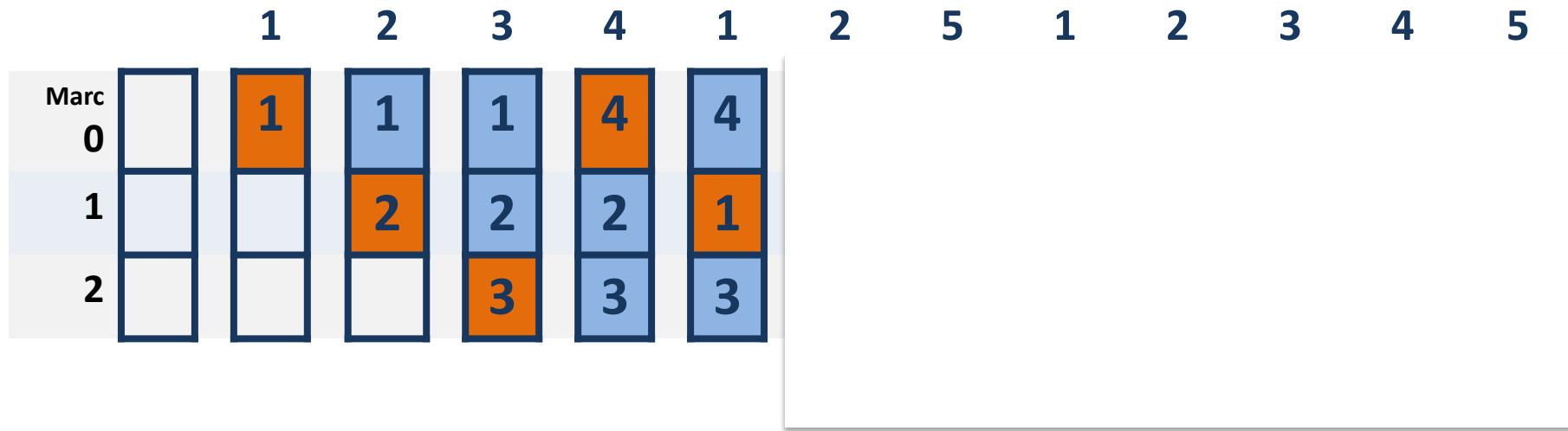
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'**anomalia de Belady**



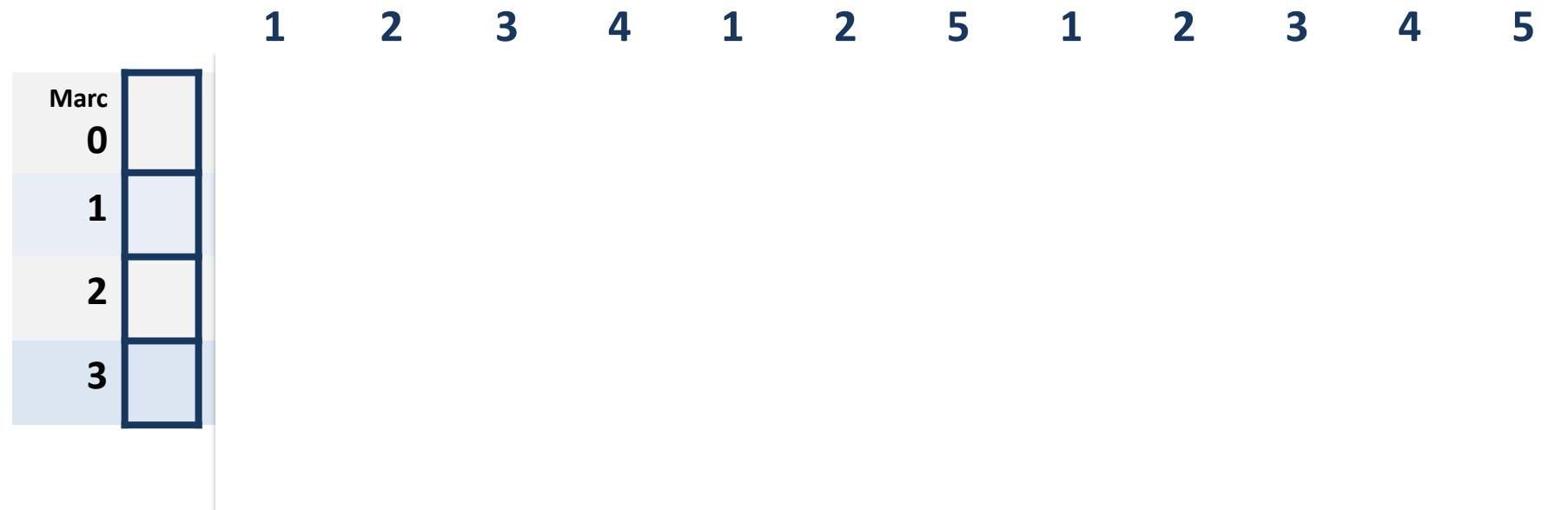
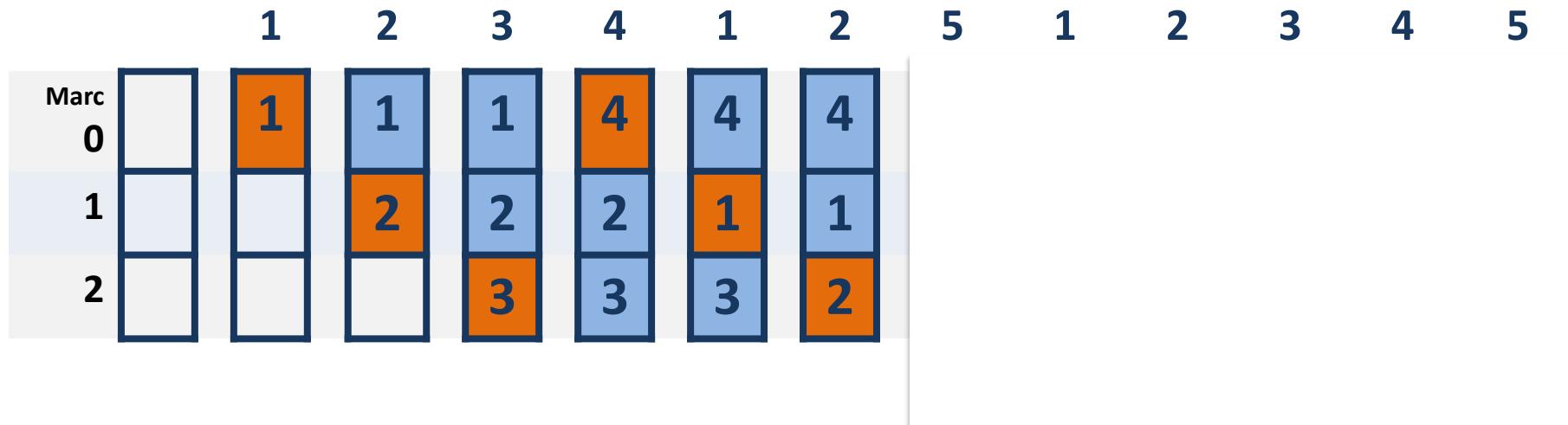
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'**anomalia de Belady**



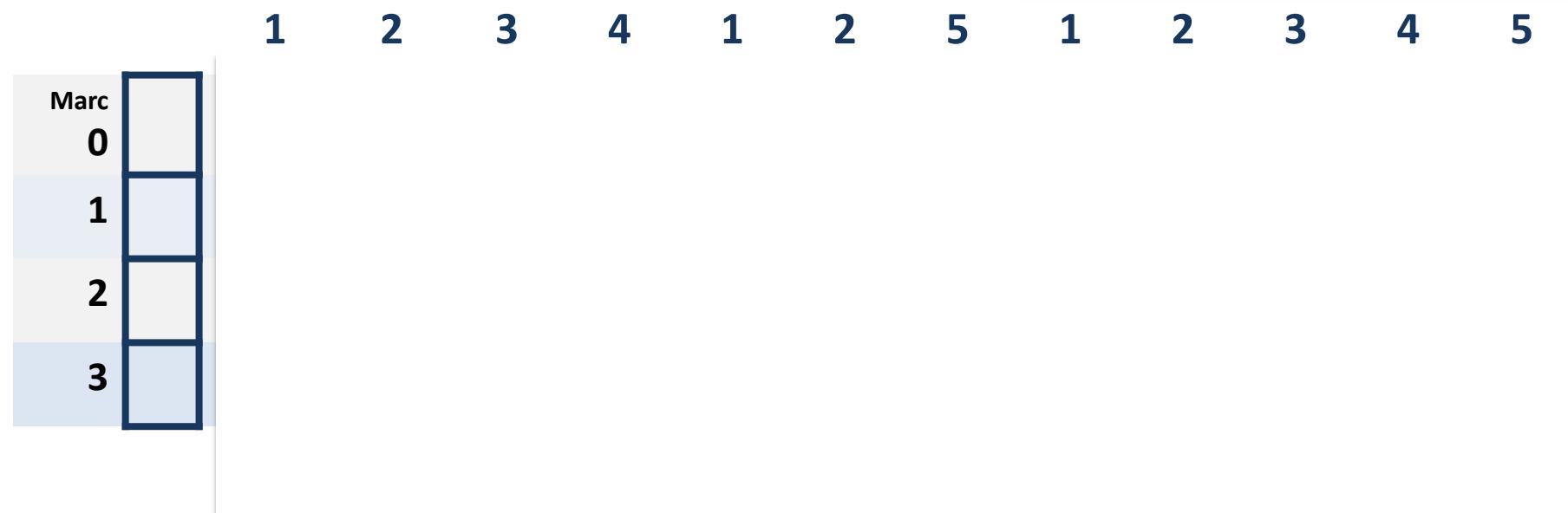
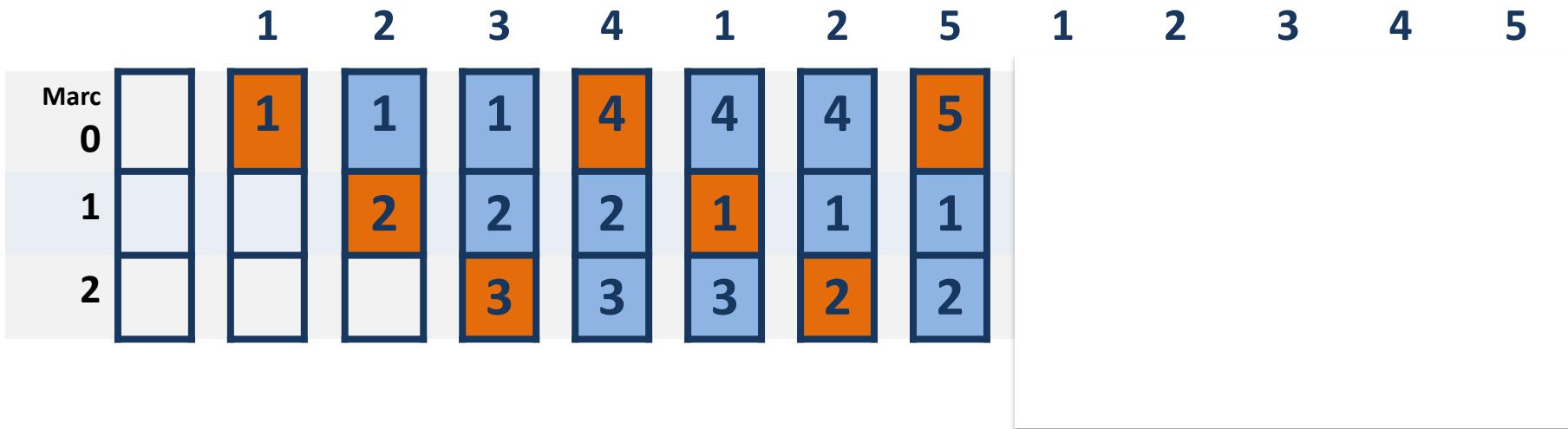
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'**anomalia de Belady**



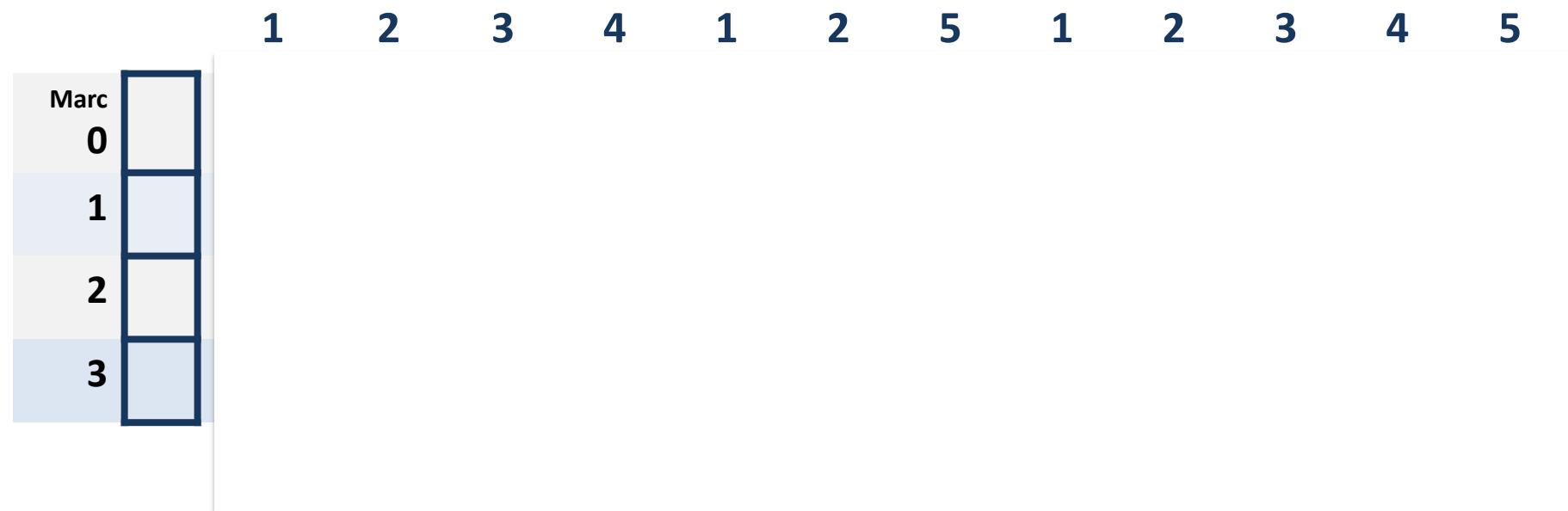
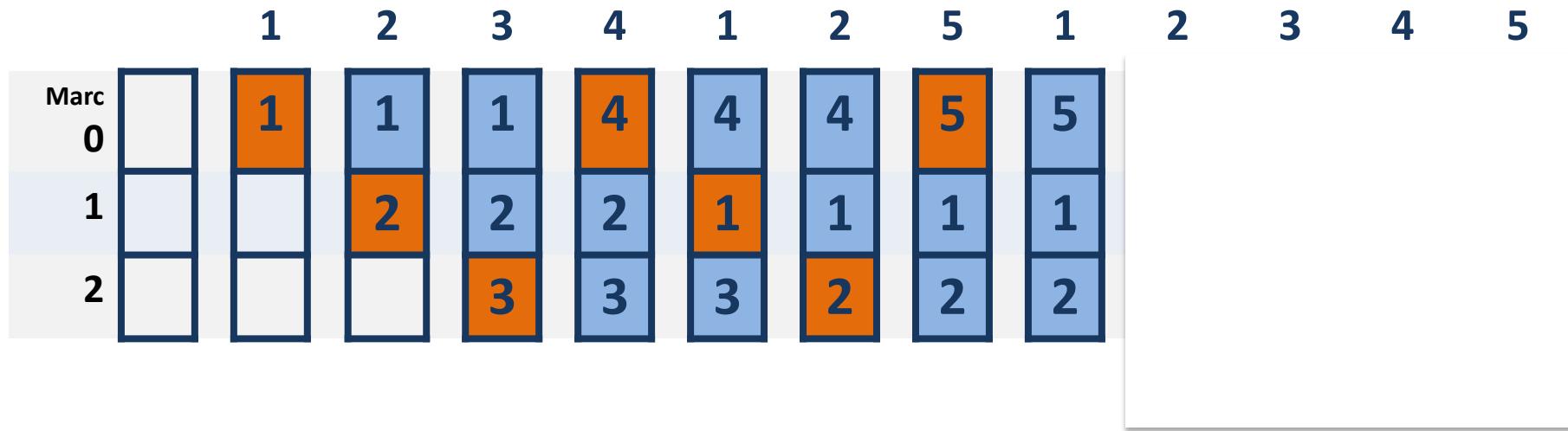
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'**anomalia de Belady**



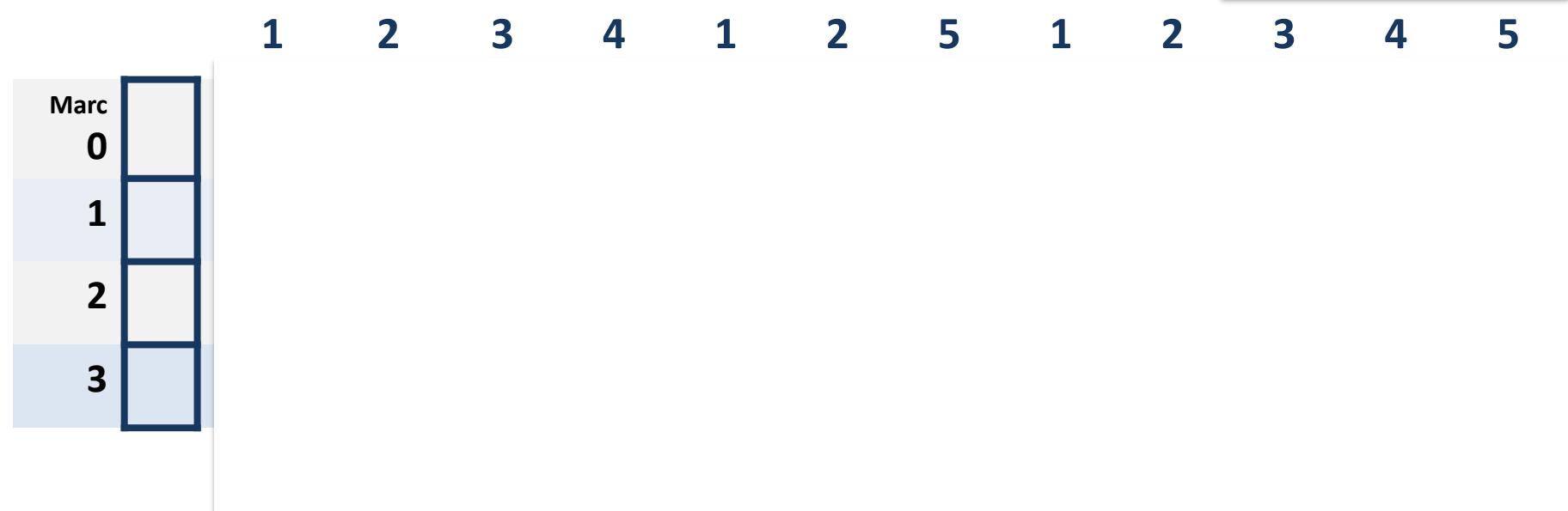
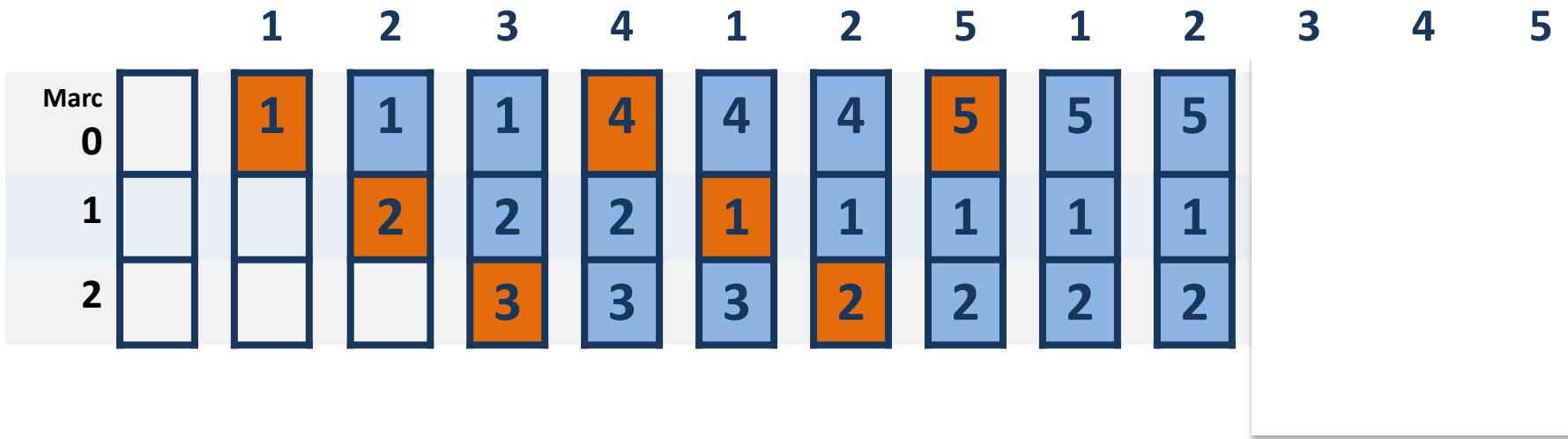
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'**anomalia de Belady**



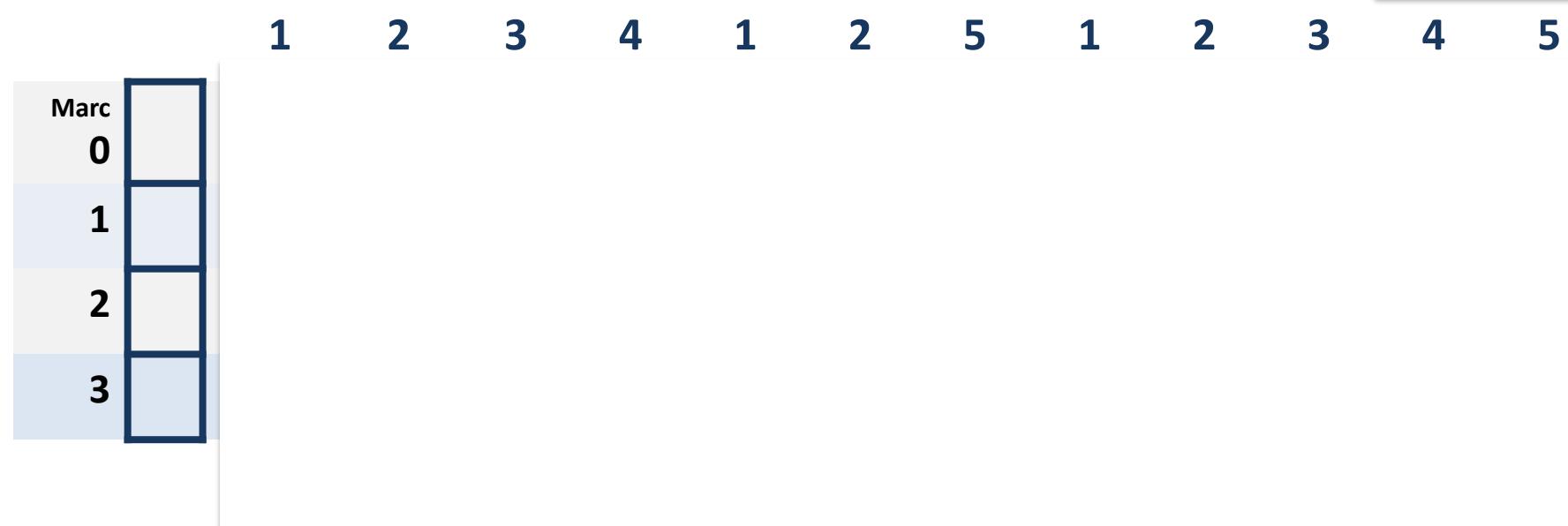
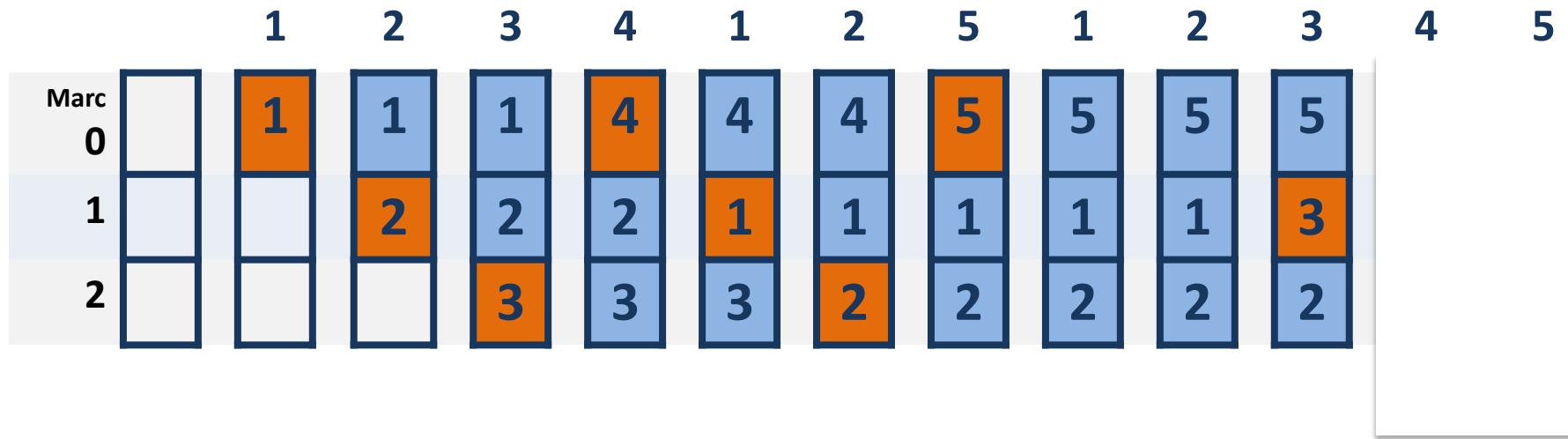
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'**anomalia de Belady**



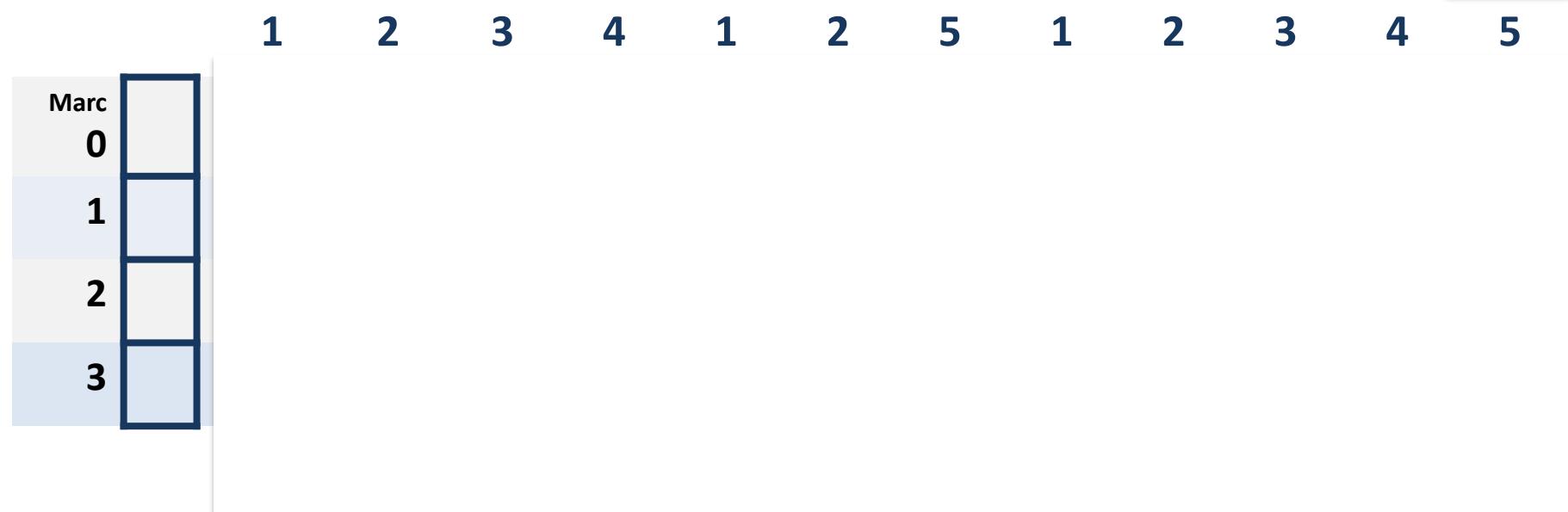
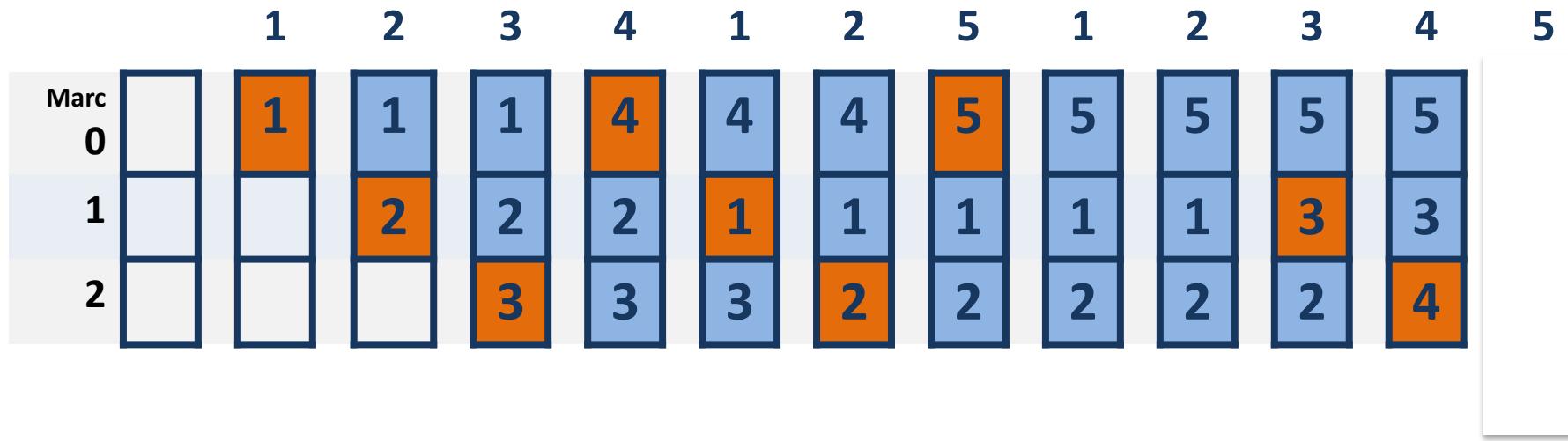
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'**anomalia de Belady**



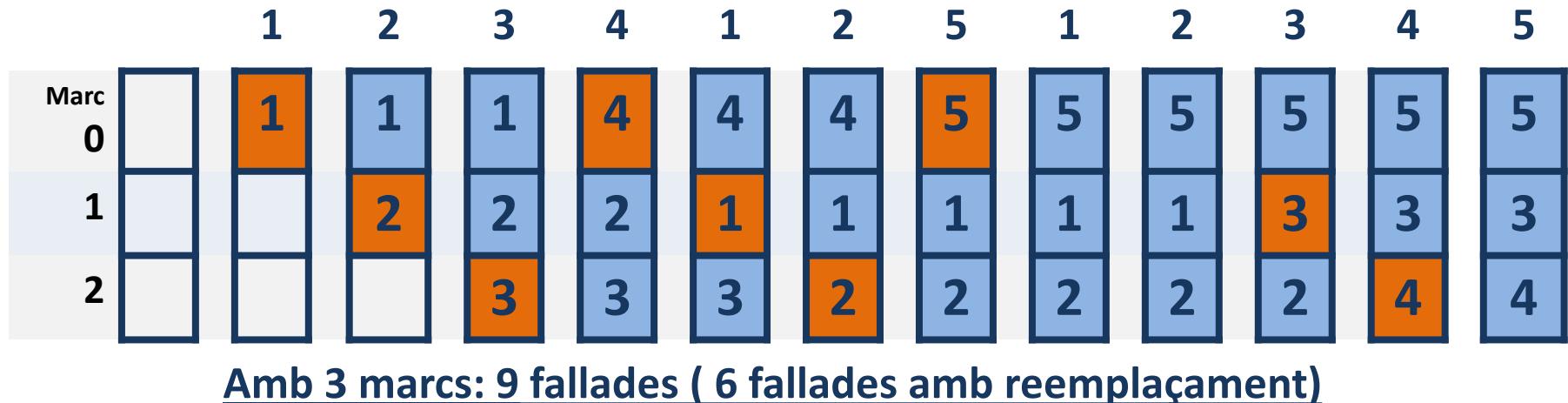
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'**anomalia de Belady**

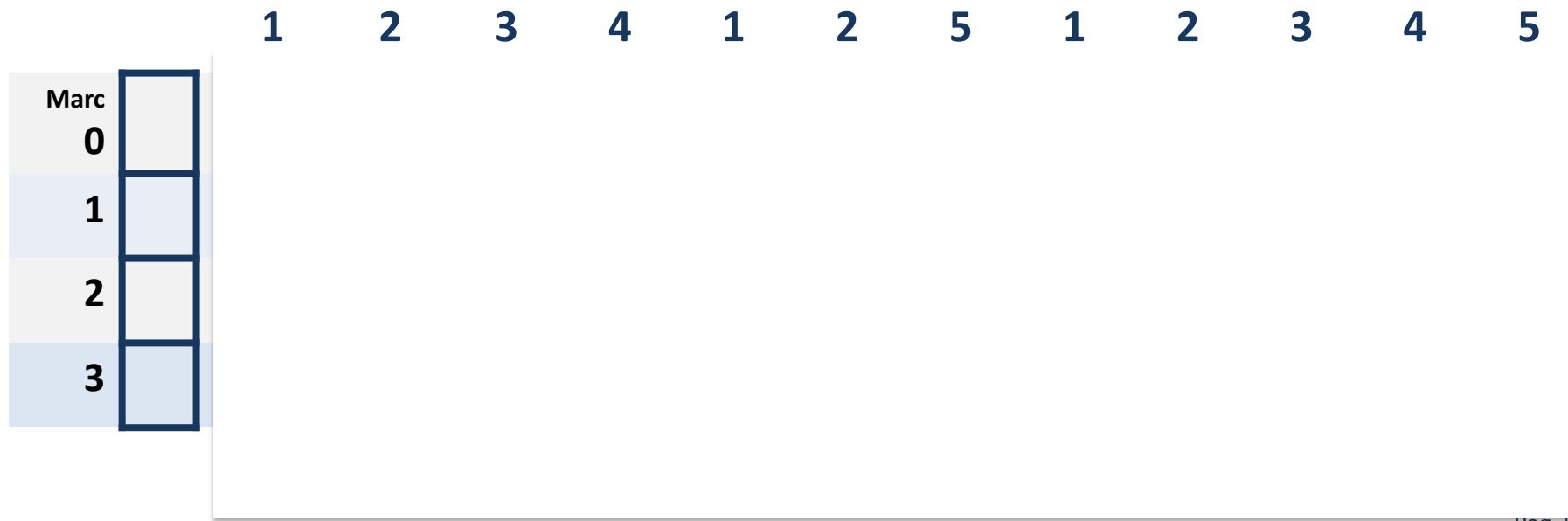


Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'anomalia de Belady

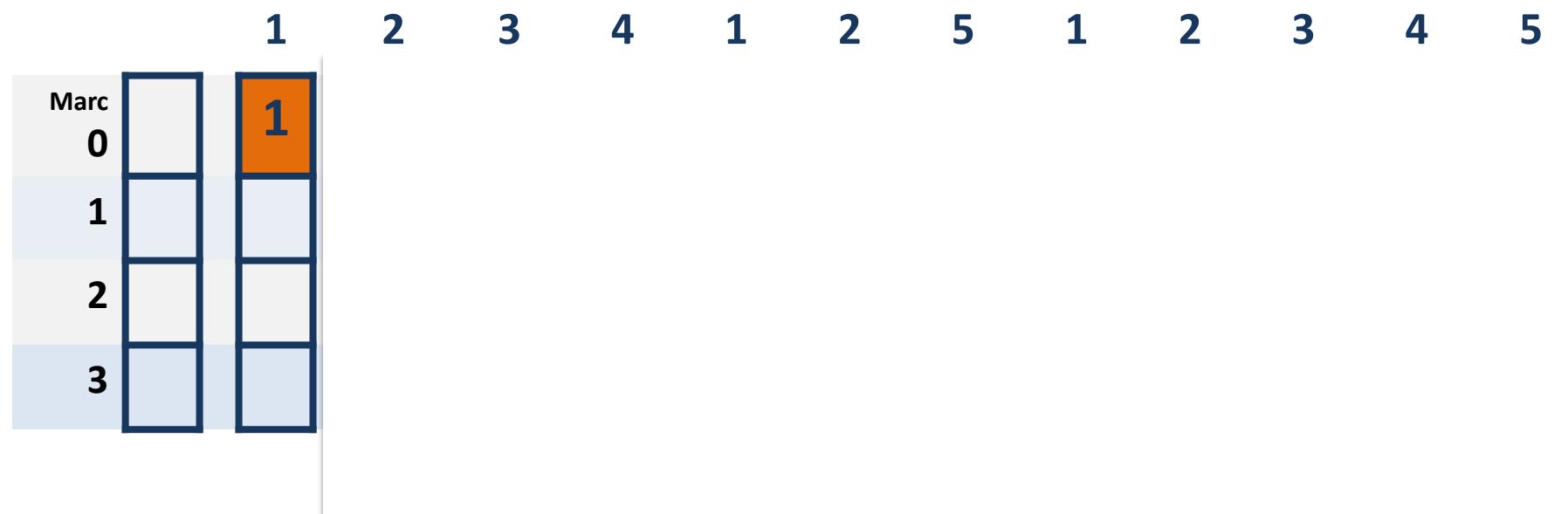
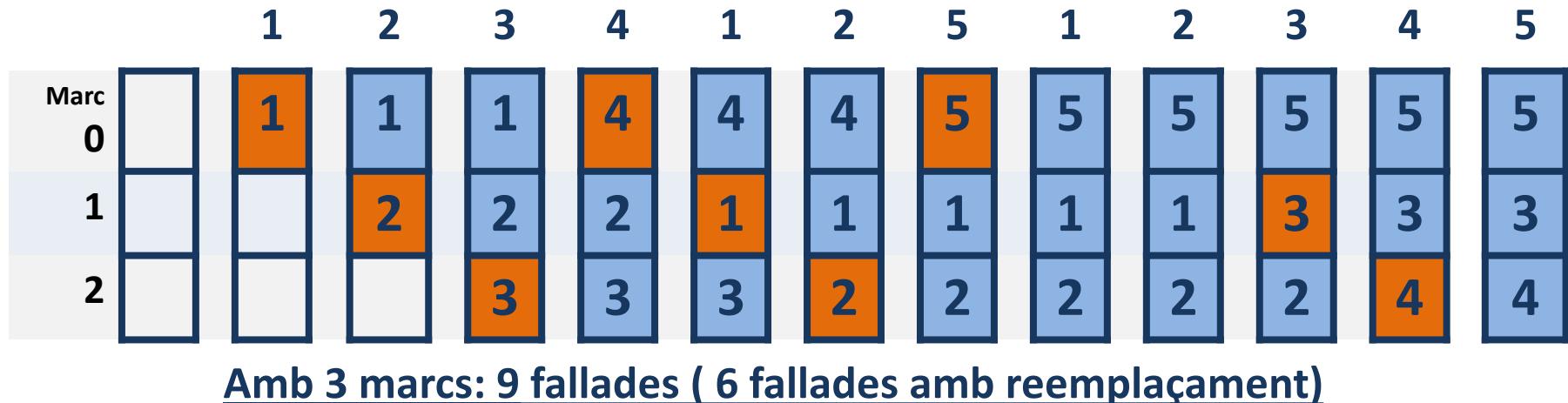


Amb 3 marcs: 9 fallades (6 fallades amb reemplaçament)



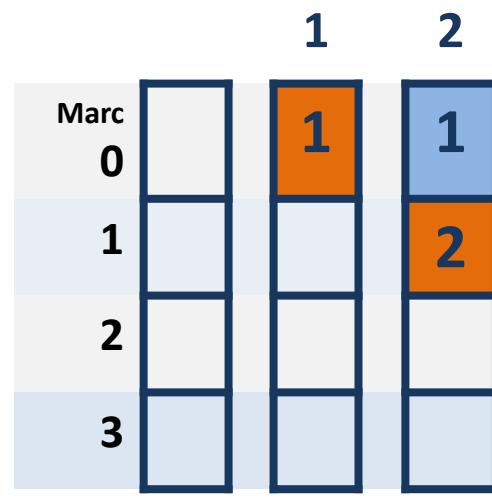
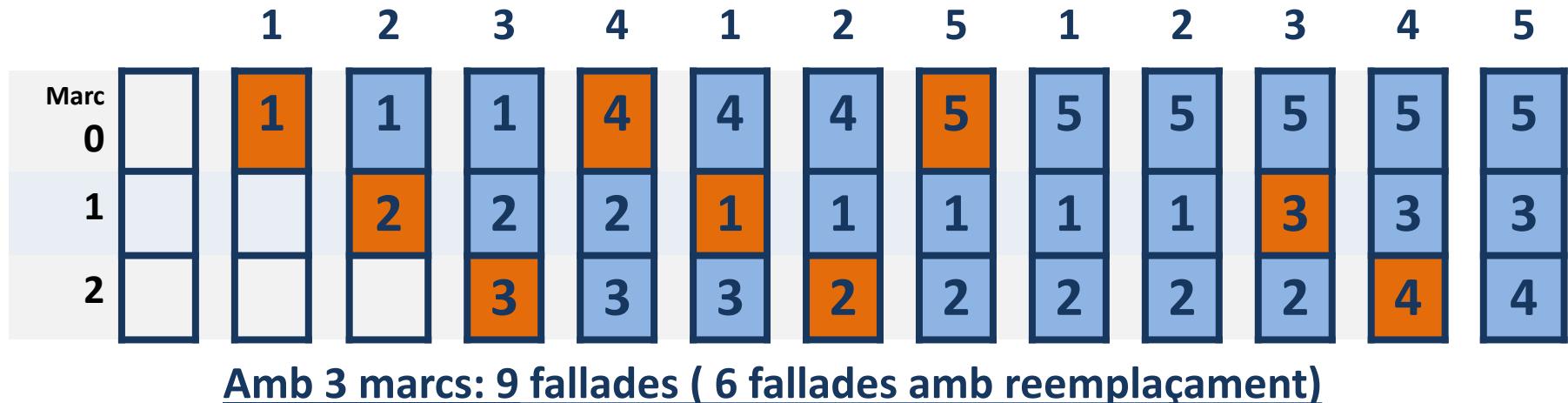
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'anomalia de Belady



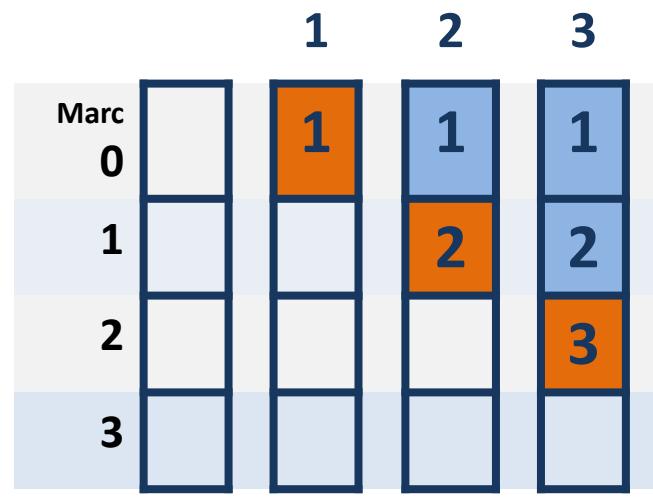
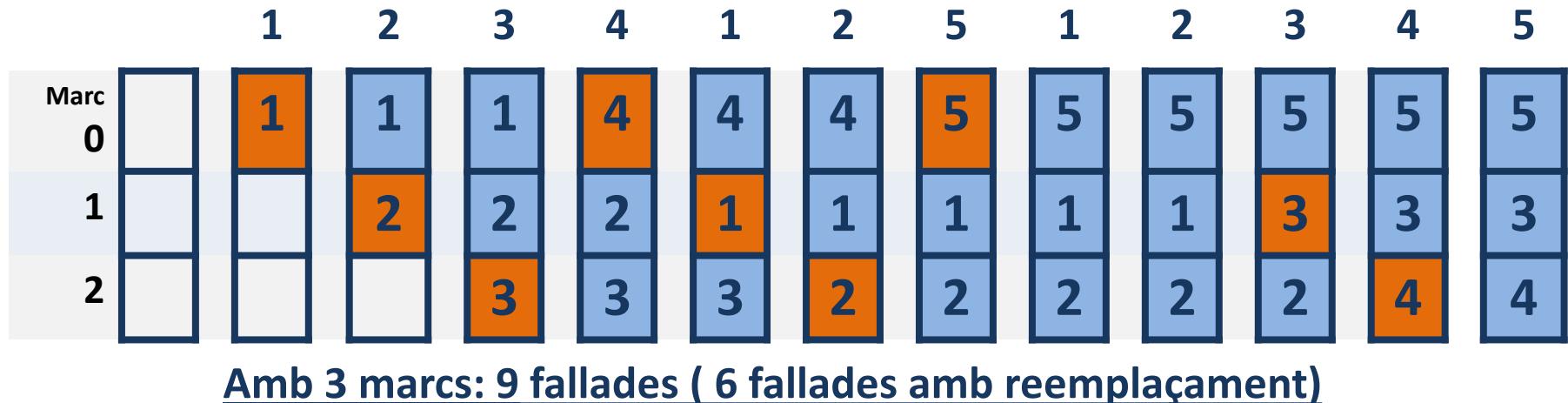
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'**anomalia de Belady**



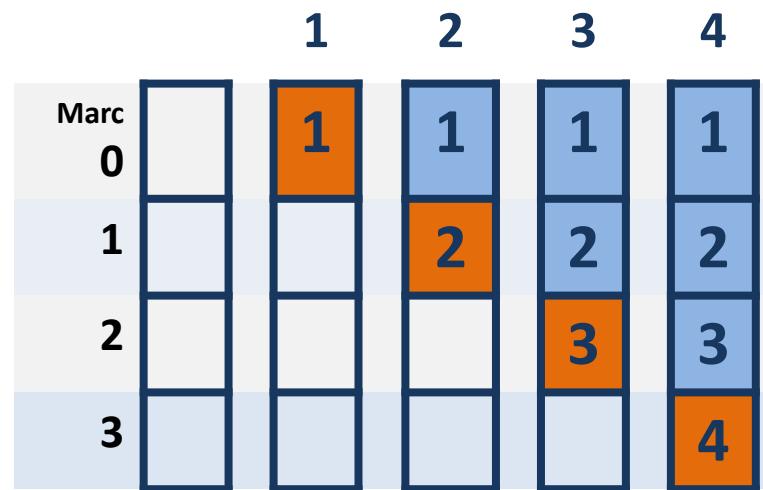
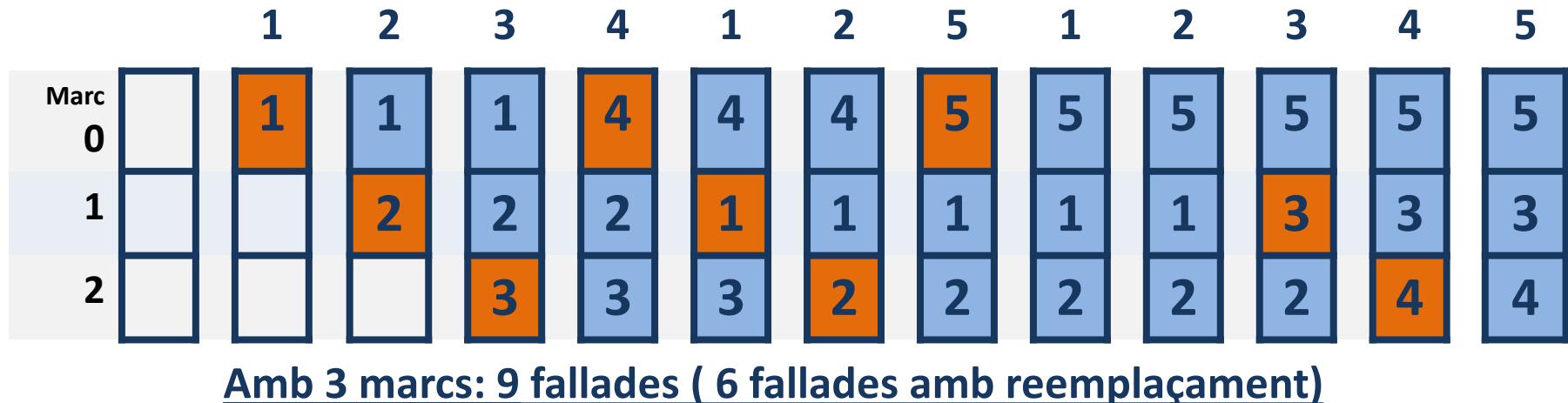
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'anomalia de Belady



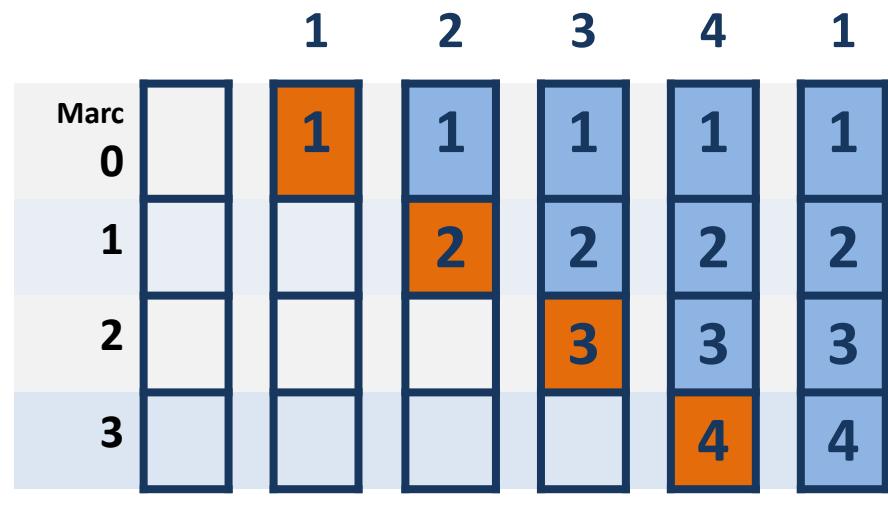
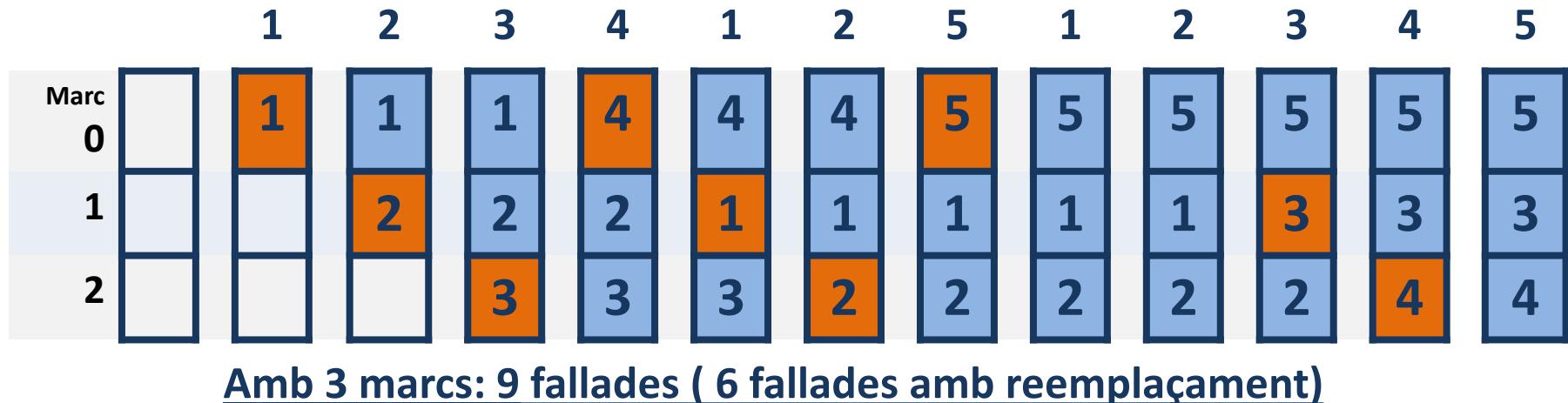
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'anomalia de Belady



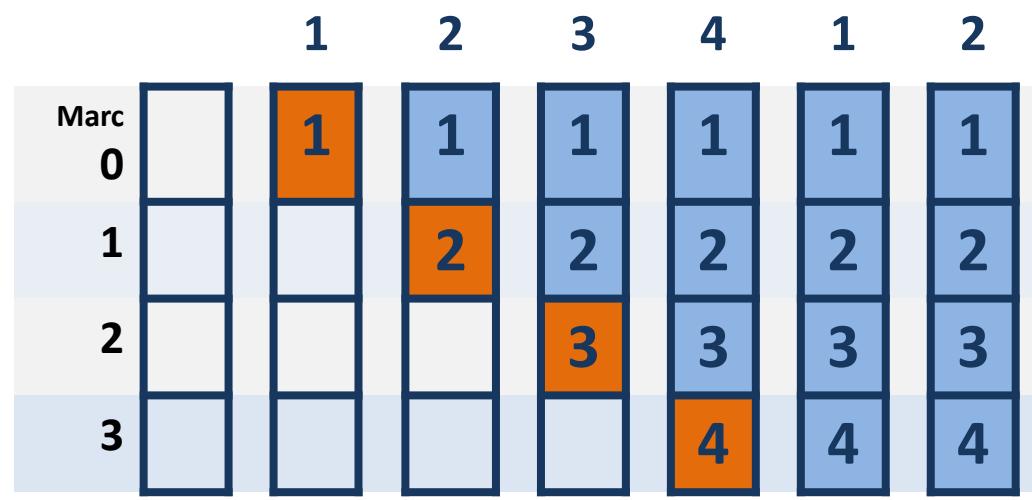
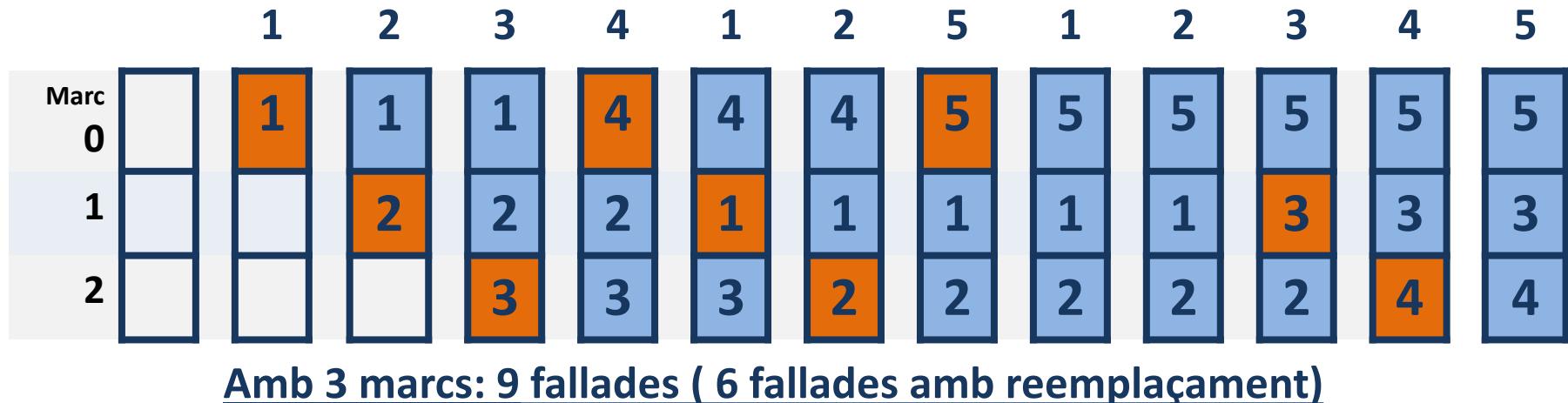
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'anomalia de Belady



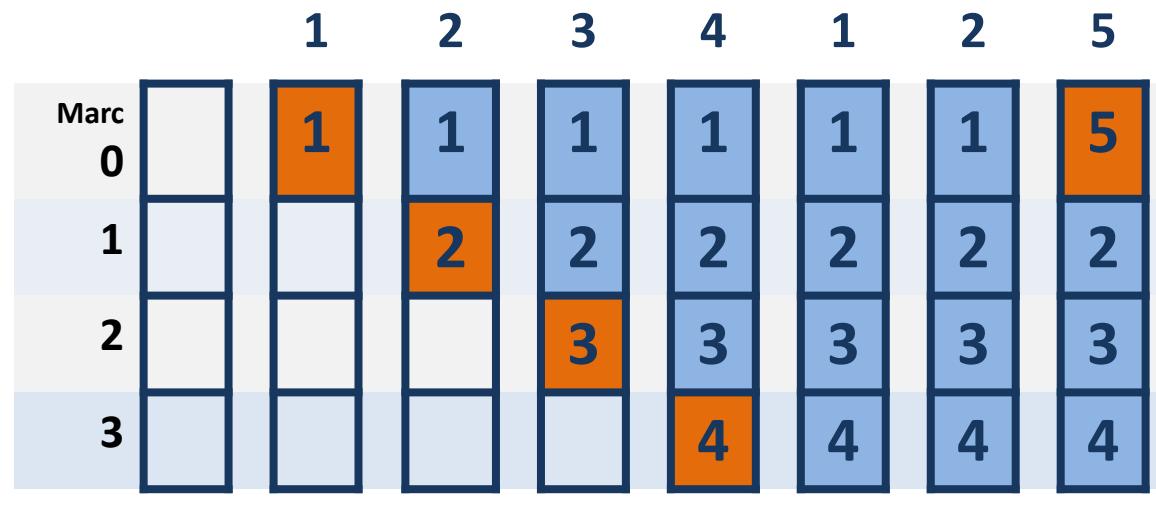
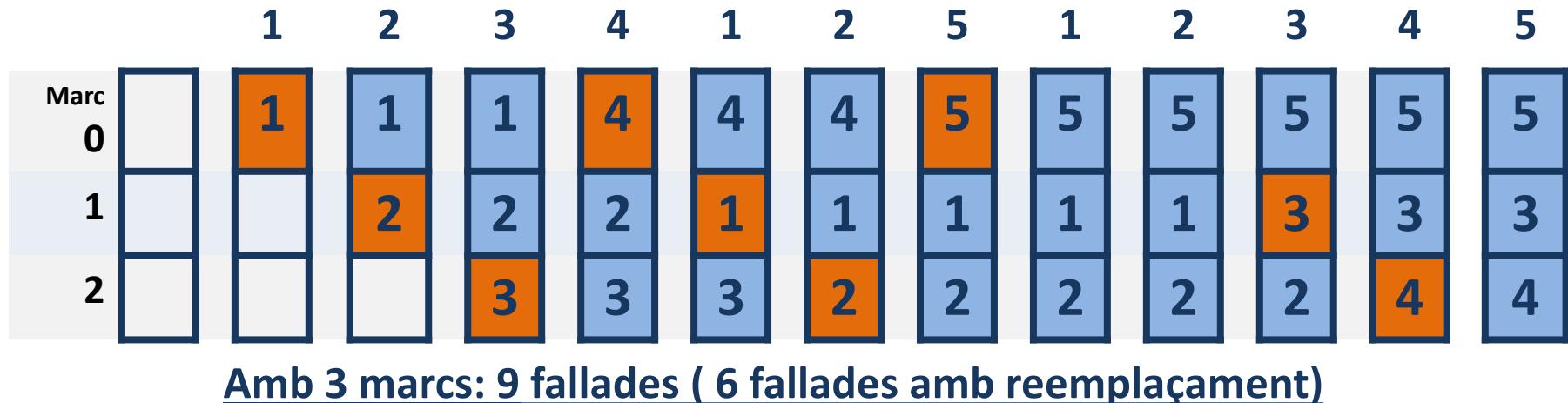
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'anomalia de Belady



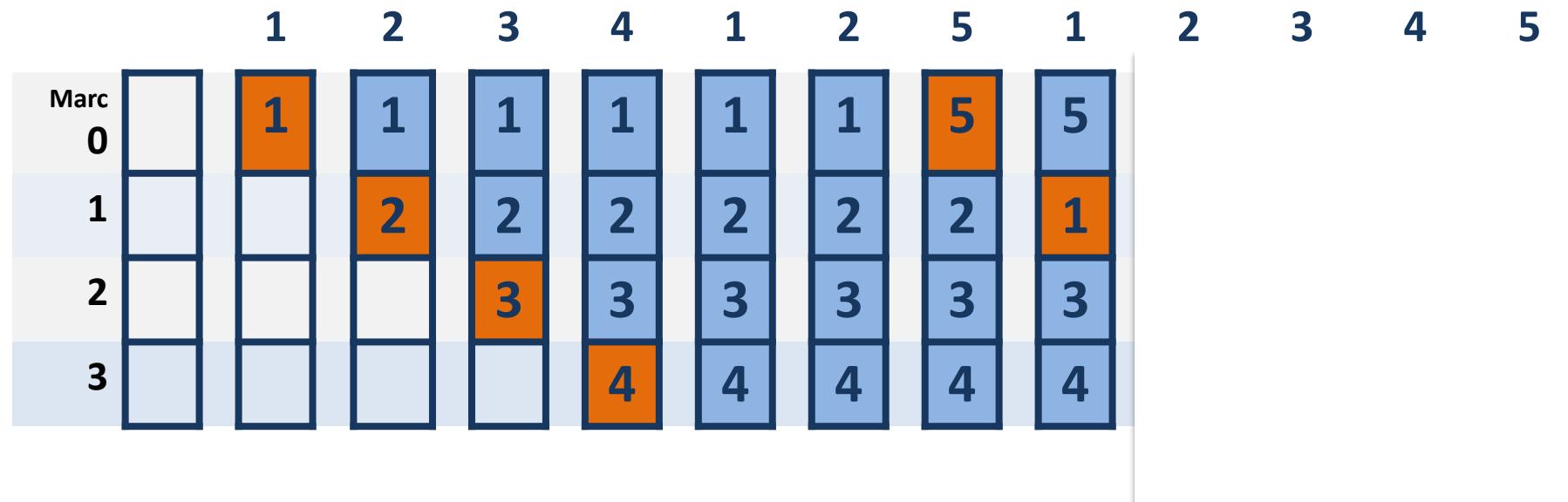
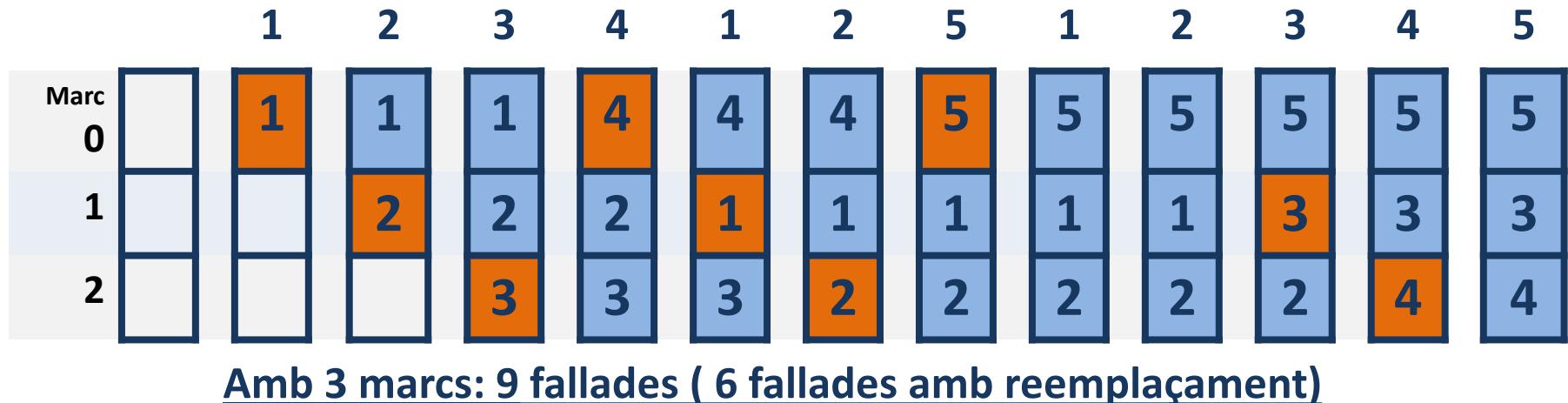
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'anomalia de Belady



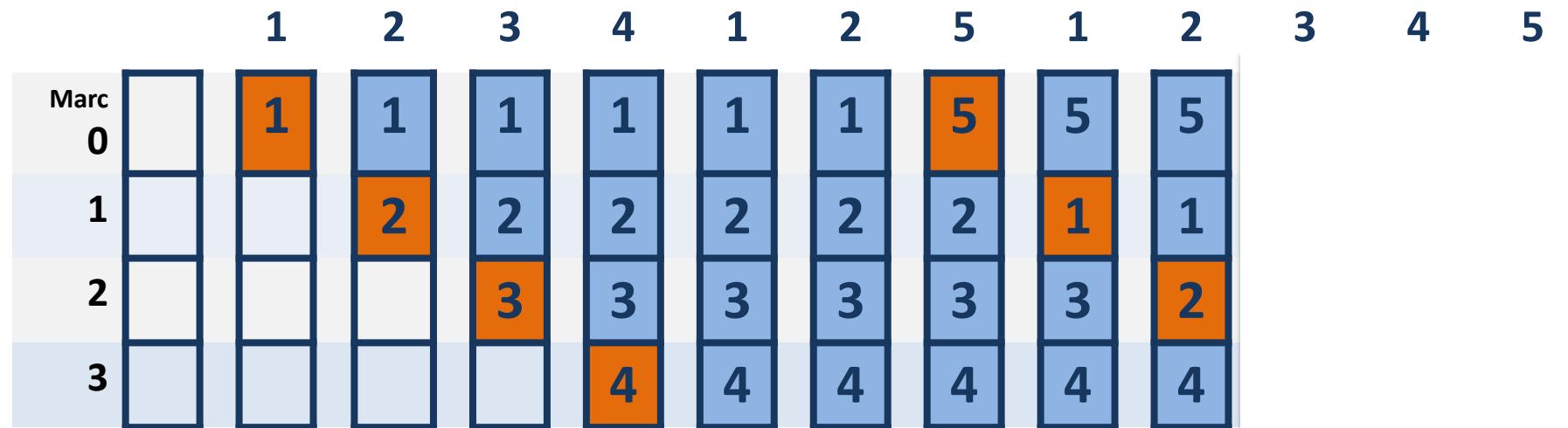
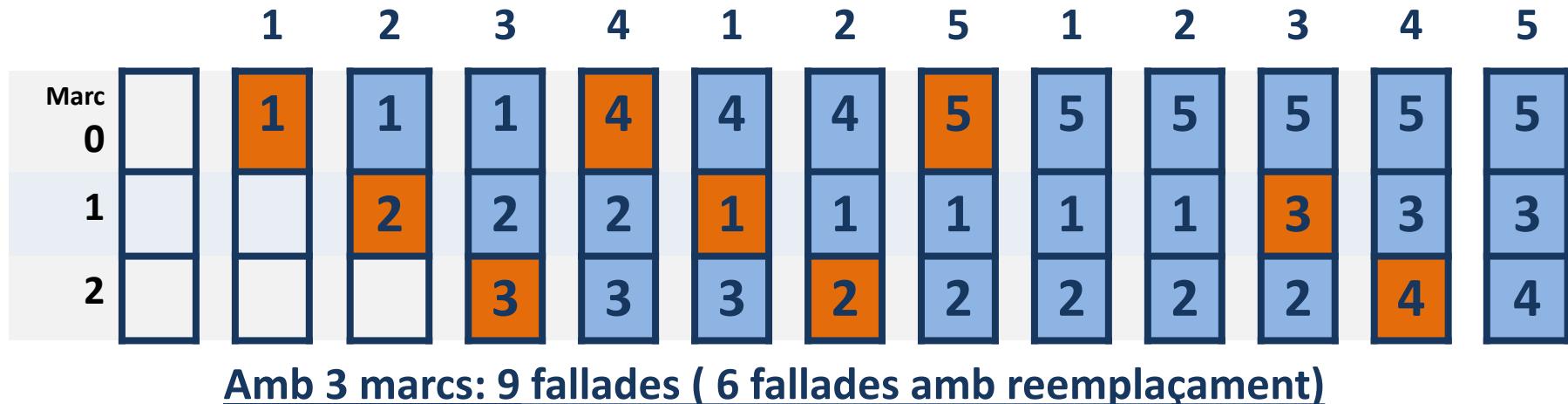
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'anomalia de Belady



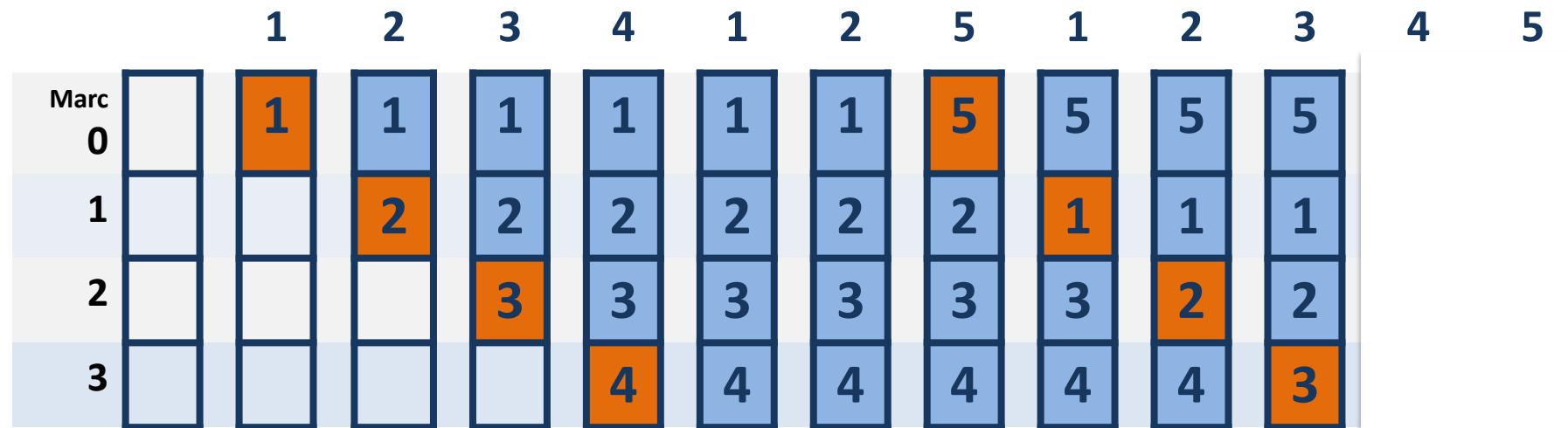
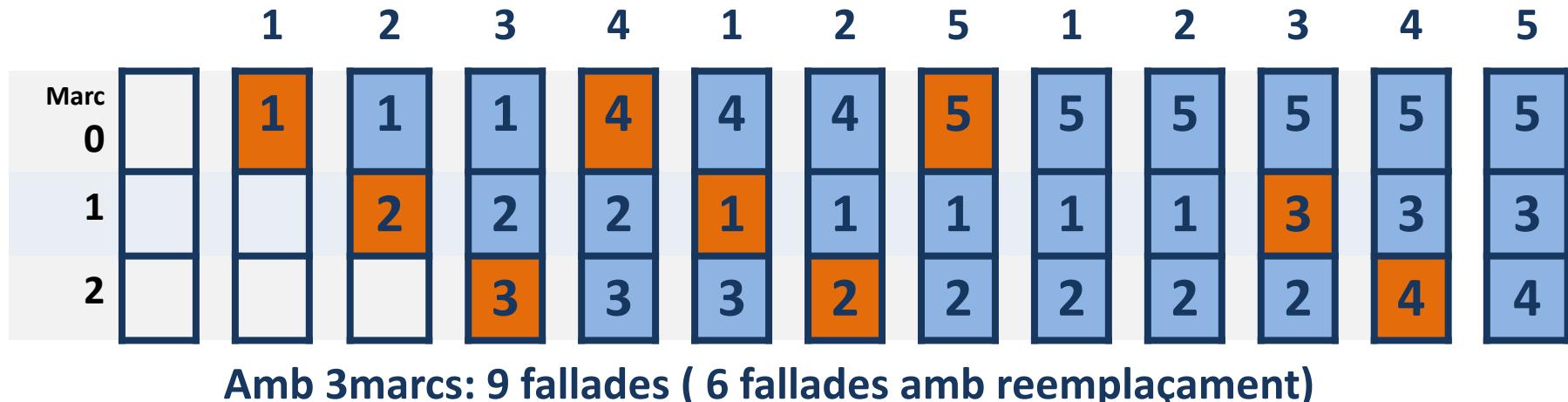
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'anomalia de Belady



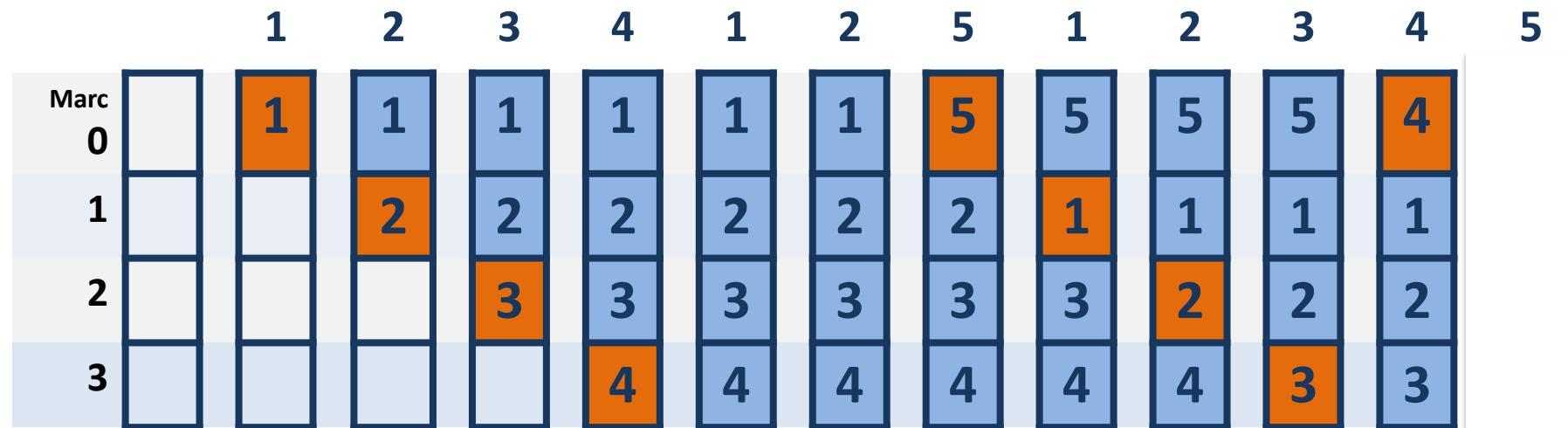
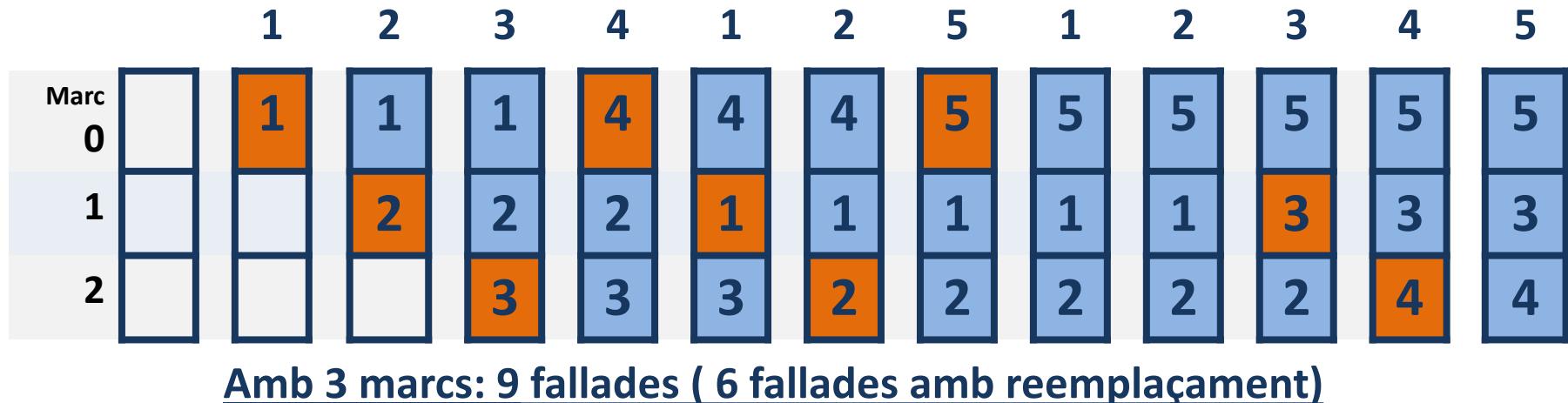
Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'anomalia de Belady



Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'anomalia de Belady



Algorisme FIFO

- Pàgina víctima: Aquella que duu més temps carregada en memòria.
- Presenta l'anomalia de Belady

