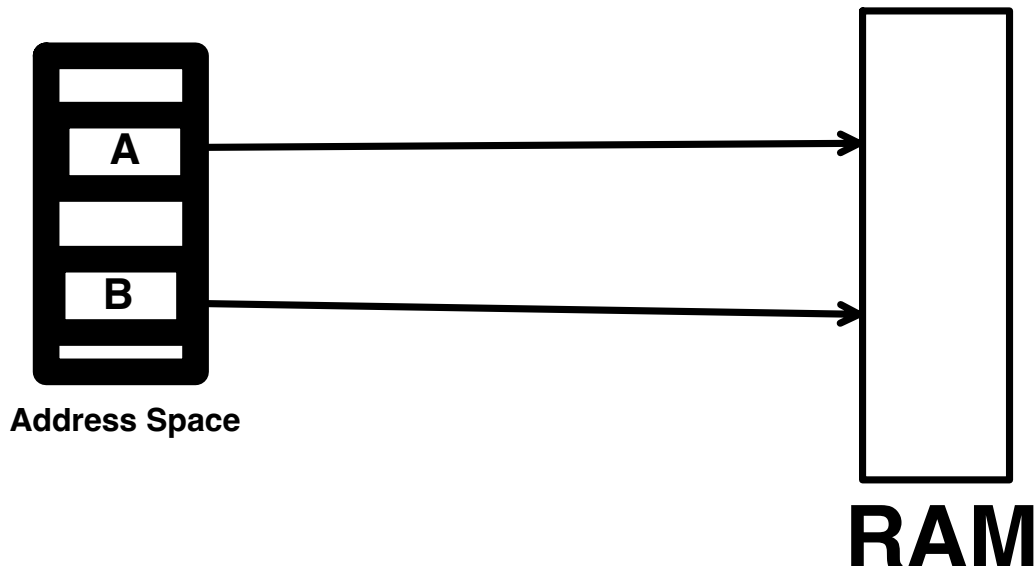


### Strategie di sostituzione delle pagine

Se noi consideriamo un address space e correntemente, di questo address space, abbiamo in memoria RAM una pagina A e una pagina B, e tutto il resto è fuori dalla memoria di lavoro, quello che dobbiamo cercare di fare è di identificare le pagine giuste da mantenere all'interno della RAM, e quindi identificare questa pagina A e questa pagina B come le pagine che "correttamente" ci permettono di far sì che l'applicazione possa realmente eseguire utilizzando informazioni che adesso sono solo in quelle pagine.



La scelta di quali pagine mantenere all'interno della RAM ovviamente viene fatta dal sistema operativo. Il fatto di selezionare A e B e scartare il resto (quindi mantenere il resto fuori ORA) è una scelta che il sistema operativo deve fare.

In tutto questo entrano in gioco vari aspetti che, a livello di sistema operativo devono essere coinvolti per andare a gestire la memoria virtuale e in particolare per andare a capire quali siano le pagine che debbano essere realmente sostituite all'interno della RAM e quindi quali siano le pagine che devono essere mantenute all'interno della RAM per una specifica applicazione nel momento in cui quest'ultima è attiva e i suoi thread stanno eseguendo delle attività.

Si tratta di stabilire quale pagina presente in memoria debba essere sostituita per far posto ad una nuova pagina da caricare

Gli aspetti coinvolti sono:

-**Gestione del Resident Set:** e quindi gestione della quantità di pagine fisiche che noi diamo in uso alle varie applicazioni.  
-Nell'insieme delle pagine che noi manteniamo all'interno della RAM, se dobbiamo eliminare qualcuna di queste per far spazio ad altre informazioni che devono essere riportate in RAM/materializzate ora in RAM, dobbiamo avere la possibilità di scegliere qual è la VITTIMA giusta.  
-Se una pagina è stata modificata nel momento in cui questa era rappresentata all'interno della RAM, e poi ovviamente dobbiamo portare questa pagina fuori dalla RAM, dobbiamo essere in grado di identificare il fatto che stiamo swappando questa pagina fuori dalla RAM, ma questa è aggiornata rispetto ad un eventuale copia che potevamo già avere sull'area di swap, da cui avevamo prelevato le informazioni per effettuare lo swap in della pagina stessa.  
Abbiamo portato in memoria qualcosa che poi abbiamo aggiornato in memoria. Questo aggiornamento lo dobbiamo riportare fuori memoria, perché altrimenti quando riportiamo questa pagina all'interno della memoria chiaramente lo stato d'esercizio di questa pagina non è coerente.

#### Aspetti coinvolti

- resident set management
  1. se l'insieme delle pagine da considerare per la sostituzione deve essere limitato alle pagine del processo che causa il page fault oppure si devono considerare pagine in un qualsiasi frame
  2. il numero di frame che devono essere allocati per ciascun processo
- nell'insieme di pagine considerate, quale “**vittima**” scegliere
- tener traccia di eventuali modifiche apportate alla pagina attualmente presente nel frame da sovrascrivere (gestione della coerenza della copia della pagina presente su hard-drive)

Se la sostituzione delle informazioni - quindi quando dobbiamo eliminare qualcosa dalla RAM per far spazio a qualcos'altro - deve avvenire, deve considerare soltanto i frame che noi abbiamo attribuito ad uno specifico processo (quindi quelli che gli abbiamo dato in uso a lui) oppure in qualche modo qualsiasi frame.  
Ovviamente il numero di frame che noi diamo in uso ad uno specifico processo chiaramente è importante per determinare se, quella quantità di memoria fisica che stiamo dando in uso per materializzare le pagine di quello specifico address space, è sufficiente per ottenere prestazioni adeguate.

Abbiamo anche delle restrizioni quando facciamo la gestione della memoria virtuale.

Una di queste restrizioni si chiama blocco di frame.

Alcuni dei FRAME della memoria RAM possono essere bloccati, ciò significa dire che il suo contenuto non è correntemente sostituibile.

Quindi non è vero che noi possiamo prendere una pagina della RAM, quindi il contenuto di un frame (una pagina fisica), e metterci dentro qualcos'altro, quindi magari caricarci un'altra pagina logica. Quindi non possiamo portare il contenuto di questo frame sull'area di swap con lo scopo di sostituire un'altra pagina logica, che magari deve essere swappata all'interno della RAM oppure materializzata ORA per la prima volta.

Questa cosa non la possiamo attuare se il frame è **BLOCCATO**.

Il blocco di frame è importantissimo per andare a gestire l'I/O asincrono, quest'ultimo prevede che noi utilizziamo alcuni dispositivi DMA per andare a far sì che un dispositivo D possa andare direttamente nel frame nella RAM a scaricare delle informazioni, e quindi se noi stiamo utilizzando questa zona della RAM come un BUFFER in cui noi andiamo a scaricare queste informazioni, e poi mentre il dispositivo D sta lavorando decidiamo di piazzare in questa zona la pagina di un altro processo che magari ha deciso di utilizzare il suo address space per certi scopi, chiaramente questa è un'operazione non corretta: permettiamo al dispositivo D di aggiornare dei dati di un address space relativo ad un processo che non c'entra niente rispetto a quando abbiamo deciso di utilizzare questo frame per eseguire questa attività di I/O asincrona, in particolare utilizzando tecniche DMA.

Il KERNEL di un sistema operativo tipicamente ha tutte le sue pagine che sono esattamente bloccate in memoria, in modo tale che qualora queste pagine siano anche utilizzate per le operazioni di I/O asincrono, il contenuto di quelle pagine viene utilizzato esattamente per questo scopo, e quindi non per ospitare pagine degli address space dei processi (parte user dei processi) e sostituire anche le informazioni all'interno di questi frame.

Sui sistemi operativi convenzionali abbiamo la possibilità di utilizzare un cosiddetto "LOCK BIT" che ci permette, dato un frame che ospita una certa pagina logica P, di mantenere questa pagina logica bloccata all'interno di questo frame. Possiamo richiedere che alcune pagine dell'address space siano bloccate all'interno della memoria fisica, e quindi non possono essere soggette alla scelta di una vittima da sostituire.

---

## Strategie di sostituzione delle pagine

Si tratta di stabilire quale pagina presente in memoria debba essere sostituita per far posto ad una nuova pagina da caricare

### Aspetti coinvolti

- resident set management
  1. se l'insieme delle pagine da considerare per la sostituzione deve essere limitato alle pagine del processo che causa il page fault oppure si devono considerare pagine in un qualsiasi frame
  2. il numero di frame che devono essere allocati per ciascun processo
- nell'insieme di pagine considerate, quale “**vittima**” scegliere
- tener traccia di eventuali modifiche apportate alla pagina attualmente presente nel frame da sovrascrivere (gestione della coerenza della copia della pagina presente su hard-drive)