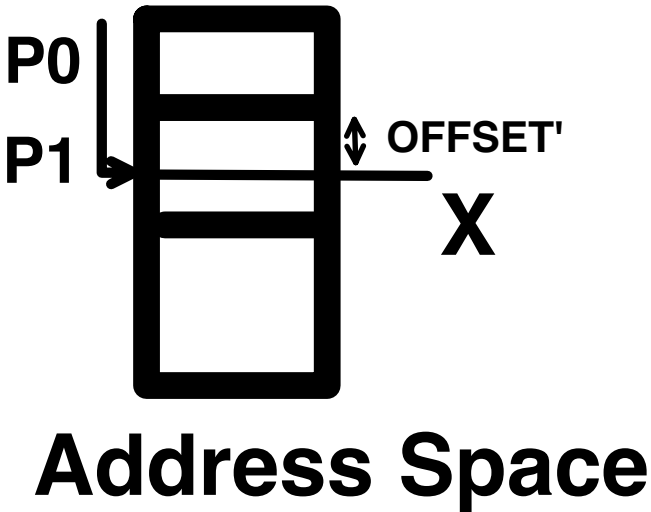


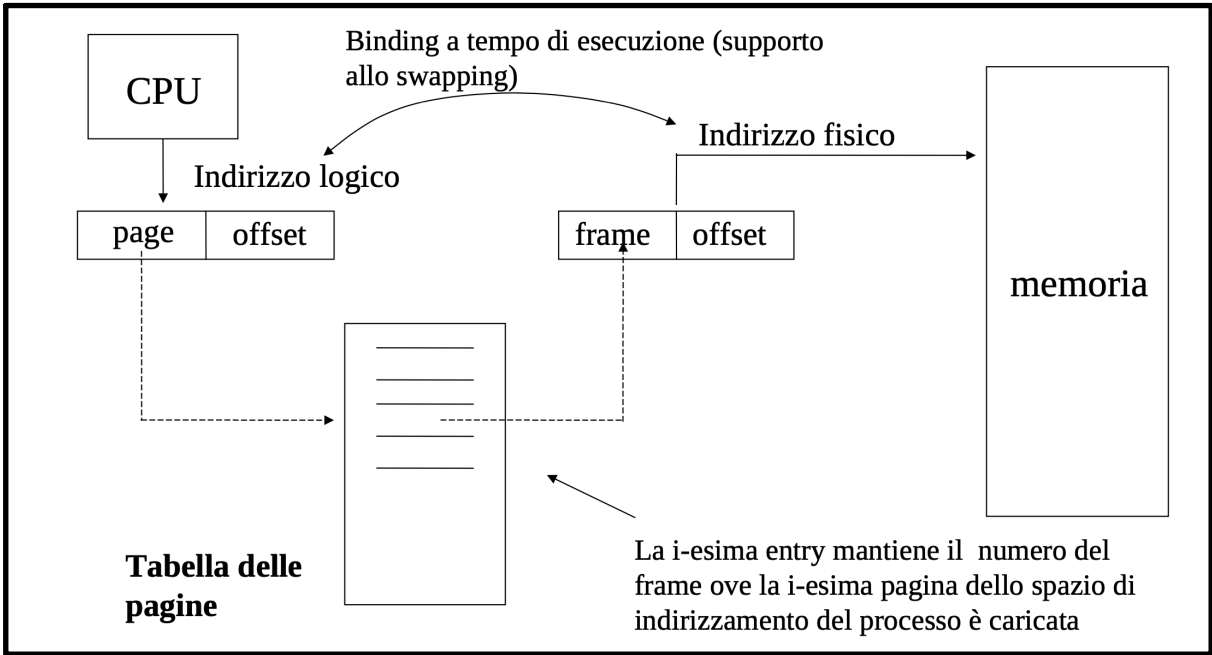
Supporti per la paginazione - tabella delle pagine

Chiaramente la MMU è molto più complessa.
Quando si esprime un "indirizzo logico" nell'address space, si esprime esattamente questa situazione qui:
Vogliamo di specificare di voler accedere esattamente in un punto preciso X dell'address space, e questo è un offset che in qualche modo stiamo esprimendo, è un indirizzo logico. Questo offset casca all'interno di una specifica pagina.



Noi stiamo cascando nella pagina P1 ad un certo OFFSET' di questa pagina!
Quindi quando si esprime, da un'istruzione, la volontà di accedere/riferire qualcosa all'interno di un address space, noi dobbiamo identificare la pagina a cui dobbiamo accedere, perché poi dobbiamo identificare la pagina fisica a cui dobbiamo accedere, ma all'interno della pagina fisica dobbiamo accedere applicando esattamente OFFSET', che è lo spiazzamento logico all'interno di una specifica pagina: il mapping delle pagine logiche sulle pagine fisiche mantiene gli offset all'interno delle pagine.

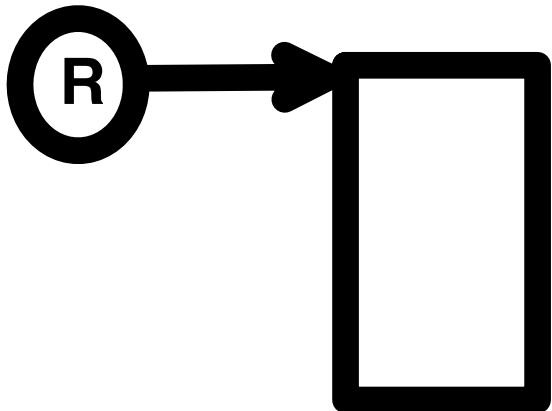
Questo implica dire che, quando noi esprimiamo un indirizzo logico in realtà tutti i bit dell'indirizzo logico vengono ad essere utilizzati: la parte alta per stabilire il numero della pagina che stiamo utilizzando, e all'interno della pagina ci muoviamo con la parte meno significativa, ossia con l'offset.



Chiaramente ad ogni accesso che noi facciamo verso la memoria, l'indirizzo logico deve essere mappato nel corrispettivo indirizzo fisico. Quindi dobbiamo identificare il FRAME (la posizione in memoria di questo frame) ed indicare l'offset della pagina (che è lo stesso applicato per l'indirizzo logico). Per effettuare questa associazione tra indirizzo logico ed indirizzo fisico dobbiamo passare attraverso una struttura complessa tale per cui, se le pagine logiche sono scorrelate in termini di posizionamento di memoria fisica - quindi non è vero che se conosciamo la posizione di uno conosciamo la posizione degli altri -, per ciascuno di queste dobbiamo mantenere in una MMU un'informazione che va ad indicare qual è esattamente il frame in RAM che corrisponde esattamente a questo oggetto.

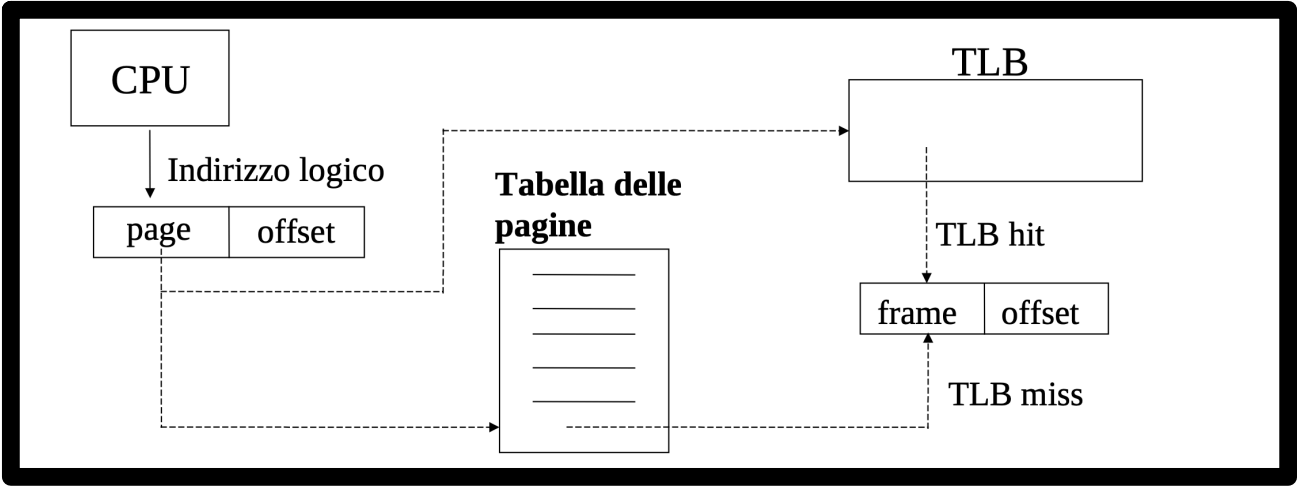
Questa corrispondenza non viene mantenuta all'interno di una MMU, viene parzialmente mantenuta in MMU.
Ma la struttura dati basica che noi abbiamo è una TABELLA DELLE PAGINE.

La tabella delle pagine è una struttura dati residente in RAM e viene mandata in setup quando viene effettuato il setup dell'address space di un processo.
E all'interno si va a scrivere, per ogni pagina logica, qual è il corrispettivo FRAME che la sta ospitando.
E ogni volta che si cerca di accedere bisogna consultare questa page table.
È il processore che consulta la tabella delle pagine, tramite il sotto-sistema di controllo.
Per cui all'interno del processore abbiamo un registro che, per il processo che correntemente è in CPU, identifica la posizione di memoria fisica della tabella delle pagine, in modo tale che il sotto-sistema di controllo possa andare a reperire la corrispettiva informazione per determinare l'indirizzo fisico (FRAME) a cui bisogna accedere tramite l'OFFSET per eseguire l'accesso all'address space.
La tabella delle pagine di un'applicazione è in un punto della RAM ad un certo indirizzo fisico, e l'informazione di qual è l'indirizzo fisico noi la carichiamo all'interno della CPU - nel momento in cui mandiamo in CPU questa applicazione, quindi lo fa il S.O - utilizzando un registro R che è di PUNTAMENTO alla tabella delle pagine correnti.



RAM

Ovviamente ci sono situazioni in cui abbiamo delle ottimizzazioni, per cui in particolare all'interno di un'architettura moderna abbiamo un componente hardware facente parte della zona MMU, che è una cache, che si chiama TLB (Translation-Lookaside-Buffer), e data una pagina logica nell'address space il processore consultando questa cache va a determinare il numero del frame a cui dobbiamo accedere.



Non c'è bisogno di passare nella tabella delle pagine, quindi di attendere il tempo di accesso alla RAM. Abbiamo questo mapping tra indirizzo logico e indirizzo fisico all'interno di una entry di questa cache ospitata all'interno del nostro processore.

Se abbiamo un miss, il sotto-sistema di controllo utilizza il registro R che è presente all'interno del processore, per andare ad identificare la posizione della tabella delle pagine per reperire la entry e per risolvere l'indirizzo, e infine per fare il fill dell'area del TLB.

Quando cambiamo processo (NON THREAD: 2 THREAD DELLO STESSO PROCESSO SONO NELLO STESSO AB) su un sistema operativo moderno, e abbiamo quindi un content switch da P1 a P2, dobbiamo resettare questa cache TLB e cambiare il puntamento del registro R verso un'altra tabella (PAGE TABLE) in RAM, quella relativa al processo P2.

Chiaramente se abbiamo una hit all'interno del TLB, ci permette di evitare le latenze per utilizzare la page table.

- per accelerare il binding viene utilizzato un traduttore hardware tra numero di pagina e frame
- questo è denominato TLB (Translation-Lookaside-Buffer)
- il TLB è una cache che mantiene associazioni tra numero di pagina e frame per il processo correntemente in esecuzione su una data unità di processamento