

Quand'è che noi possiamo SWAPPARE fuori un'applicazione dalla memoria?

L'applicazione esegue, viene chiamato il kernel, e viene chiamato un servizio bloccante. L'applicazione ha il suo address space all'interno della RAM. Quindi se il thread è bloccato e questa applicazione è tale per cui non possiamo lavorarci sopra, possiamo prendere l'address space in RAM e portarlo su un hard drive, liberando la zona della RAM dove lui era presente.

Ma questa cosa la possiamo fare soltanto quando c'è la completa INATTIVITÀ delle attività di quella zona della RAM (AB).

In una zona della RAM noi ci arriviamo sia con la CPU, perché magari eseguiamo qualche istruzione e qualche accesso ai dati di questo address space, che con un DMA.

In particolare un DMAC che può essere programmato per andare a lavorare in una specifica zona della RAM.

I DMAC sono interessanti perché vengono ad essere utilizzati quando i thread delle applicazioni / processi, chiamano i servizi di I/O.

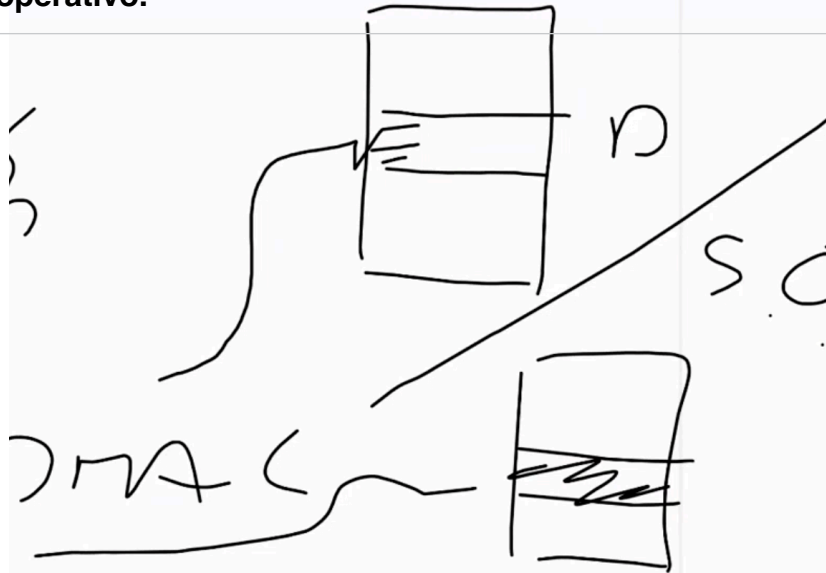
Quando andiamo su un file system e i dati non ci stanno in buffer cache e quindi il thread deve essere bloccato, il blocco deve avvenire fino a quando noi non preleviamo dati da un hard drive, e lo facciamo utilizzando approcci di tipo DMA. Programmiamo il DMAC per prendere un blocco dell'hard drive e caricarlo in una zona della memoria RAM.

Quindi se un processo chiama un servizio del Kernel per andare in I/O, e questo processo sta utilizzando la RAM, e all'interno del suo address space abbiamo la sezione .data, il DMAC non può essere programmato per scaricare direttamente informazioni all'interno della sezione .data, non possiamo direttamente far vedere questi dati che andiamo ad acquisire utilizzando i DMAC all'interno di questo AB.

Quindi il DMAC deve essere programmato per andare a scaricare dati all'interno di un'altra zona della RAM che è una zona riservata per il S.O, in particolare per il kernel. Soltanto quando i dati sono presenti in quell'altra zona, soltanto a valle di questo, si potrà effettuare la copia di questa memoria quando eventualmente il processo P è stato riportato in RAM in una specifica zona della memoria.

Questo è importante per far capire che su un sistema moderno, quando noi abbiamo lo swap out e lo swap in, questa cosa può essere fatta soltanto nel momento in cui abbiamo la completa inattività (anche l'inattività dei componenti fisici all'interno dell'AB) dello spazio di indirizzamento, in modo tale che se carichiamo in quella zona della RAM un altro AB, quei componenti fisici che magari volevano lavorare su quella zona della RAM non vanno ad interferire con il contenuto di un altro address space.

I buffer per l'I/O asincrono (inteso per le tecniche dma) vengono allocati nella memoria riservata per il sistema operativo.



- processi in stato di blocco (attesa di evento) possono essere riversati fuori della memoria (**Swap-out**) per far posto ad altri processi
- lo swapping è in genere utilizzato in caso il sistema supporti binding a tempo di esecuzione (flessibilità sullo **Swap-in** di processi precedentemente riversati fuori dalla memoria)
- in caso di binding a tempo di compilazione o caricamento, lo Swap-in deve afferire alla stessa partizione di Swap-out

## Vincoli

- per poter effettuare Swap-out è necessario che ci sia completa inattività sullo spazio di indirizzamento (I/O asincrono non permesso)
- i buffer per l'I/O asincrono vengono quindi allocati nella memoria riservata per il sistema operativo

