

TLB

Un'implementazione software dei meccanismi di paginazione rallenterebbe di molto l'esecuzione. Per esempio, su 2 livelli (IL: 10 + 10 + 12 bit), richiede due accessi extra in memoria:

```
; r0: tabella di primo livello
; r1: indirizzo logico

; mancano eccezioni/protezioni

lsr r2, r1, #22          ; offset di primo livello
ldr r0, [r0, r2, lsl 2]  ; tabella di secondo livello

lsl r2, r1, #10
lsr r2, r2, #22          ; offset di secondo livello
ldr r0, [r0, r2, lsl 2]  ; pagina fisica

and r2, r1, #4096        ; offset di byte
ldr r0, [r0, r1]         ; dato
```

Perciò si utilizza hardware dedicato (nella MMU), in particolare il *translation lookaside buffer*, che è una cache delle associazioni tra pagine logiche e fisiche.

Performance

La località temporale è molto più forte rispetto alle cache dati/istruzioni, visto che la stessa pagina è usata per un gran numero di indirizzi (4 KB). Non ci serve la pagina seguente nell'immediato, perciò non si caricano a blocchi le pagine.

Tipicamente il TLB contiene 16-512 associazioni, ed è abbastanza piccolo da avere tempi di accesso inferiori ad un ciclo di clock. Spesso l'hit rate è > 99%.

Context switch

Potremmo svuotare il TLB, ma è più efficiente memorizzare il PID relativo a ciascuna associazione. Il SO scrive in un apposito registro il PID del processo corrente, e il TLB ignora tutte le associazioni con PID diverso.

Modifica dei permessi

Quando si *riducono* i permessi di una pagina, è necessario invalidare immediatamente l'entry della pagina nel TLB. Aumentare i permessi non è un problema: si può aggiornare il TLB quando (e se) viene generata un'eccezione per via dell'incongruenza.

Nei multiprocessori ogni core ha il suo TLB, che può essere invalidato solo dal processore associato. È necessario fare un *TLB shutdown*: il SO interrompe tutti i core con una shutdown request in modo che applichino le modifiche. È un'operazione dispendiosa, perciò spesso si aspetta di avere diverse richieste prima di eseguire lo shutdown per limitare le interruzioni.

Software-loaded TLB

Non si fa traduzione degli indirizzi in hardware: in caso di miss nel TLB viene generata un'eccezione, che il SO gestisce indicando la pagina fisica richiesta. Semplifica l'implementazione a costo della performance: un miss con traduzione hardware può costare qualche decina di cicli, con software-loaded TLB diverse centinaia/migliaia.

Deve essere fornito un meccanismo per inserire un'associazione nel TLB: le operazioni di memoria sono quindi load, store, fetch e associazione TLB.

