

Soluzioni piu' raffinate alla frammentazione : allocazione non contigua della memoria

1. PAGINAZIONE

2. SEGMENTAZIONE



1. Paginazione : idea di base

Lo spazio di indirizzi logici di un processo puo' essere non contiguo

- Si suddivide la **memoria fisica** in blocchi di taglia fissa chiamati **frame**
- Si suddivide la **memoria logica** in blocchi della stessa taglia chiamati **pagine**

Per eseguire un **processo di n pagine**, si devono trovare **n frame liberi**

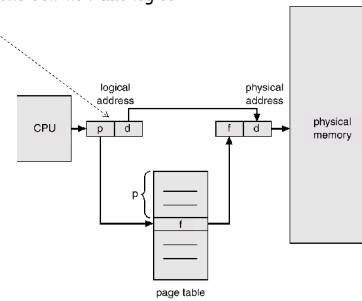
Supporto alla paginazione ...

- Tenere traccia di tutti i **frame liberi**
- Organizzare una **page table** per tradurre indirizzi logici in fisici
- Un indirizzo generato dalla CPU (logico) viene quindi suddiviso in :
 - **Numero di pagina (p)** - usato come **indice in una page table** che contiene l'indirizzo di base di ogni pagina nella memoria fisica
 - **Spiazzamento (d)** - combinato con l'indirizzo di base della pagina definisce l'indirizzo fisico di memoria

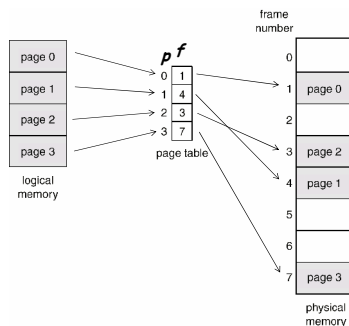


... uno schema illustrativo ...

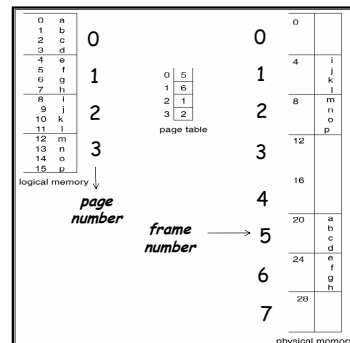
Una **taglia** di pagina **potenza di 2** agevola la suddivisione dell'indirizzo logico



... un esempio di page table ...



... ed un altro esempio ...



... e qualche considerazione

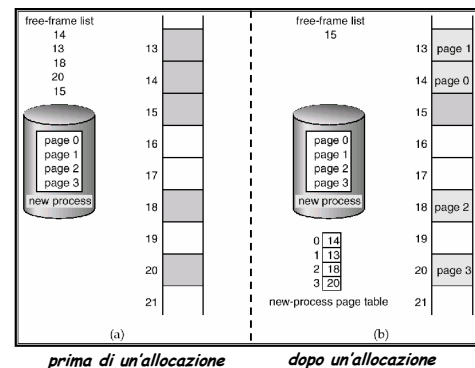
- La **dimensione fissa** dei frame evita anche il problema di **allocare blocchi** di dimensioni diverse **nel disco** quando vengono deallocati
- La **frammentazione interna** e' limitata solo all'ultima pagina allocata ad un processo
- **Dimensione** di una pagina (pagine piccole comunque generano page table con molte entries)
- **Tabella/lista dei frame** (liberi e occupati) gestita da OS

Sistemi Operativi

36

Vittorio Cortellesa, 2002-2003

Tabella frame liberi



Sistemi Operativi

37

Vittorio Cortellesa, 2002-2003

Come cambia la gestione di un processo ?!

- Una page table per ogni processo
- **Un campo in piu' nel PCB**: (il puntatore alla) page table
- **Aggiornamento dei frame** quando il processo viene scaricato o ricaricato
- **Context switching piu' oneroso** in quanto coinvolge anche la gestione delle pagine

Sistemi Operativi

38

Vittorio Cortellesa, 2002-2003

Implementazione di una page table

La page table puo' essere **tenuta in memoria** con il supporto di:

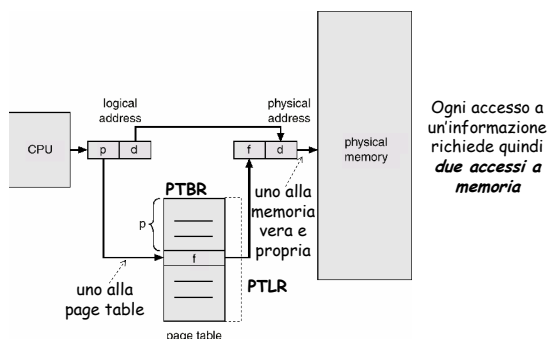
- **Page-table base register (PTBR)** punta all'**inizio** della page table
- **Page-table length register (PTLR)** indica la **taglia** della page table

Sistemi Operativi

39

Vittorio Cortellesa, 2002-2003

... e una raffigurazione



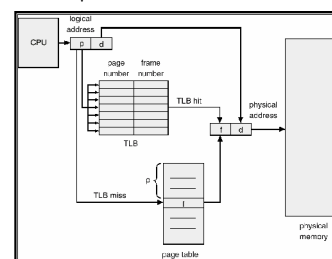
Sistemi Operativi

40

Vittorio Cortellesa, 2002-2003

Soluzione al doppio accesso: memoria associativa

Una speciale cache chiamata **translation look-aside buffers (TLBs)**



Traduzione dell'indirizzo (P, D)

Se P e' in TLBs prendi il numero di frame F

Altrimenti recupera il numero di frame F dalla page table in memoria e riportalo anche in TLBs

In caso di context switching TLBs va ripulita

Sistemi Operativi

41

Vittorio Cortellesa, 2002-2003

Tempo di accesso effettivo (EAT) con memoria associativa

- **Accesso alla cache**: ε microsecondi
- **Accesso alla memoria**: 1 microsecondo
- **Tasso di hit**: α - percentuale di volte che un numero di pagina viene trovato in TLBs (proporzionale al numero di registri)

$$EAT = (1 + \varepsilon) \alpha + (2 + \varepsilon)(1 - \alpha) = 2 + \varepsilon - \alpha$$

Provate ad assegnare dei valori per comprendere l'andamento di questa funzione!



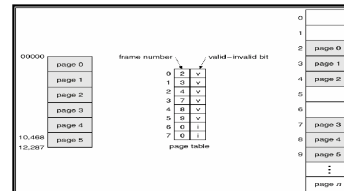
Sistemi Operativi

42

Vittorio Cortellessa, 2002-2003

Protezione della memoria paginata

- Bit di protezione possono essere associati ad ogni frame: **lettura, scrittura, esecuzione**
- Un bit di **validita'** puo' essere associato ad ogni entry di una page table:
valid indica che la pagina e' nello spazio di indirizzi logici del processo

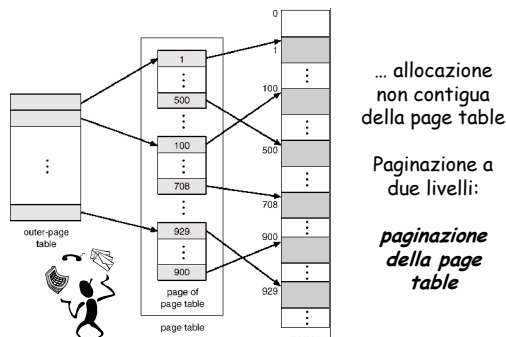


Sistemi Operativi

43

Vittorio Cortellessa, 2002-2003

E per page table troppo lunghe...



Sistemi Operativi

44

Vittorio Cortellessa, 2002-2003

... un esempio ...

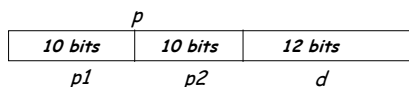
- Un indirizzo logico (su una macchina a 32-bit con una pagina di taglia 4K) viene suddiviso in:
 - **un numero di pagina** di 20 bits (p)
 - **un offset** di 12 bits (d)
- Poiche' **la page table e' paginata** (p.es. con pagine di taglia 1K), il numero di pagina viene a sua volta suddiviso in:
 - **un numero di pagina** di 10 bits (p1)
 - **un offset all'interno della pagina della page table** di 10 bits (p2)

Sistemi Operativi

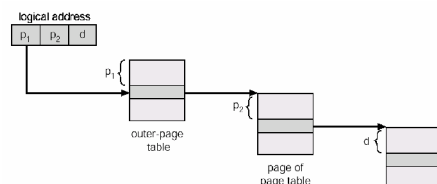
45

Vittorio Cortellessa, 2002-2003

Un indirizzo logico ha quindi il seguente formato:



dove $p1$ e' l'indice della outer page table, e $p2$ e' lo spaziamento all'interno della pagina della page table



Sistemi Operativi

46

Vittorio Cortellessa, 2002-2003

... e tempo di accesso

- Poiche' ogni livello e' memorizzato come una separata tabella in memoria, **la traduzione da logico a fisico puo' implicare 3 accessi a memoria** nel caso di paginazione della page table (i livelli possono essere anche di piu'!)
- Il **caching** potrebbe permettere di tenere un tempo di accesso ragionevole
- Un **hit ratio** del 98% porta a :

$$EAT = 0.98 \times (1 + \varepsilon) + 0.02 \times (3 + \varepsilon) = 1 + (\varepsilon + 0.04) \mu s$$

che e' un rallentamento del $(\varepsilon + 0.04)\%$ del tempo di accesso a memoria

Sistemi Operativi

47

Vittorio Cortellessa, 2002-2003