Paginazione e segmentazione

Index

- Paginazione (semplice)
- Segmentazione (semplice)
 - Indirizzi Logici

Paginazione (semplice)

La paginazione semplice in quanto tale non è stata sostanzialmente mai usata, ma è importante a livello concettuale per introdurre la memoria virtuale.

Con la paginazione sia la memoria che i processi vengono "spacchettati" in pezzetti di dimensione uguale. Ogni pezzetto del processo è chiamato **pagina**, mentre i pezzetti di memoria sono chiamati **frame**.

Ogni pagina, per essere usata, deve essere collocata in un frame ma pagine contigue di un processo possono essere messe in un qualunque frame (anche distanti)

I SO che la adottano però devono mantenere una tabella delle pagine per ogni processo che associa ogni pagina del processo al corrispettivo frame in cui si trova.

(i) Quando c'è un process switch, la tabella delle pagine del nuovo processo deve essere ricaricata ed aggiornata

A differenza di prima in cui l'hardware doveva solamente intervenire e aggiungere un offset, qui deve intervenire sulle pagine stesse, infatti un indirizzo di memoria può essere visto come un numero di pagina e uno spiazzamento al suo interno (indirizzo logico)

i≣ Example

Frame number 0	Main memory
0	
2	
3	
4	
1 2 3 4 5 6 7	
6	
7	
8	
9	
10	
11	
12	
13	
14	

Deve essere caricato un processo A che occupa 4 frame

Frame number	Main memory
0	A.0
1	A.1
2 3 4 5	A.2
3	A.3
4	
5	
6 7	
7	
8	
9	
10	
11	
12	
13	
14	

Ne arrivano altri due da 3 (B) e 4 (C) frame

Frame number 0	Main memory
0	A.0
1	A.1
2	A.2
3	A.3
4	B.0
5	B.1
6	B.2
3 4 5 6 7 8	C.0
	C.1
9	C.2
10	C.3
11	
12	
13	
14	

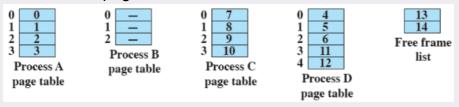
Quindi viene swappato B

Frame number	Main memory
number 0	A.0
1	A.1
2 3	A.2
3	A.3
4	
5	
4 5 6 7	
	C.0
8	C.1
9	C.2
10	C.3
11	
12	
13	
14	

E sostituito con D (con il partizionamento dinamico, non sarebbe stato possibile caricare D in memoria)

Frame number ₀	Main memory
0	A.0
1	A.1
2	A.2
3	A.3
4	D.0
5	D.1
6	D.2
1 2 3 4 5 6 7 8	C.0
	C.1
9	C.2
10	C.3
11	D.3
12	D.4
13	
14	

Tabelle delle pagine risultanti



∷ Esempio di traduzione

Supponiamo che la dimensione di una pagina sia di $100~{\rm bytes}$. Quindi la RAM dell'esempio precedente è di solo $1400~{\rm bytes}$

Inoltre i processi A, B, C, D richiedono solo 400, 300, 400 e 500 bytes rispettivamente (comprensivi di codice - program, dati globali ed heap - data e stack delle chiamate).

Nelle istruzioni dei processi, i riferimenti alla RAM sono relativi all'inizio del processo (quindi, ad esempio, per D ci saranno riferimenti compresi nell'intervallo [0,499])

Supponiamo ora che attualmente il processo D sia in esecuzione, e che occorra eseguire l'istruzione j 343 (vale lo stesso anche per istruzioni di load o store, anche se devono passare per registri)

Ovviamente non si tratta dell'indirizzo 343 della RAM: lì c'è il processo A. Bisogna capire in quale pagina di D si trova 343: basta fare 343 div 100 = 3

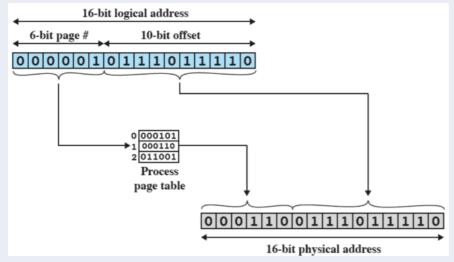
Poi occorre guardare la tabella delle pagine di D: la pagina 3 corrisponde al frame di RAM numero 11

Il frame 11 ha indirizzi che vanno da 1100 a 1199: qual è il numero giusto? Basta

fare 343 mod 100 = 43 quindi il 44-esimo byte L'indirizzo vero è pertanto $11\cdot 100 + 43 = 1143$

(i) Info

Per ogni processo, il numero di pagine è al più il numero di frames (non sarà più vero con la memria virtuale)

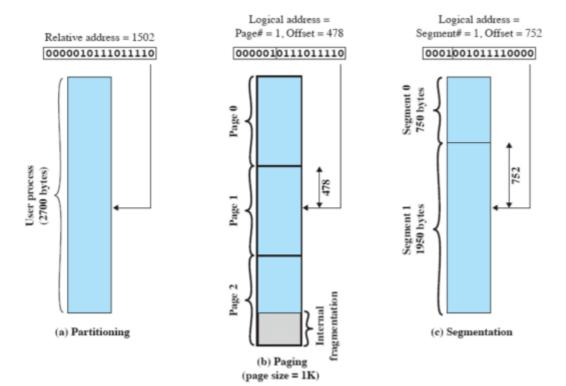


Per ottenere l'indirizzo vero dunque punto alla pagina formata dai $6 \, \mathrm{bit}$ più significativi, controllando la corrispondenza con il frame, e utilizzo i restanti $10 \, \mathrm{bit}$ (ogni pagina è grande $2^{10} \, \mathrm{bit}$) come offset all'interno della pagina

Segmentazione (semplice)

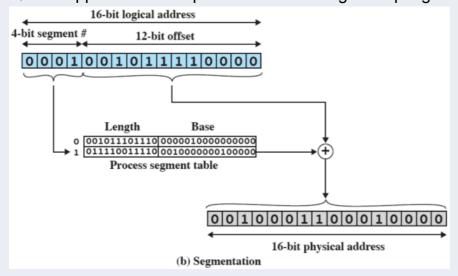
La differenza tra paginazione e segmentazione sta nel fatto che nella paginazione le pagine sono tutte di ugual dimensione, mentre i segmenti hanno **lunghezza variabile**. In questo risulta simile al partizionamento dinamico ma è il programmatore a decidere come deve essere segmentato il processo (tipicamente viene fatto un segmento per il codice sorgente, uno per i dati condivisi e uno per lo stack delle chiamate)

Indirizzi Logici



(i) Info

Qui si suppone che non possano esserci segmenti più grandi di $2^{12}\ \mathrm{bytes}$



In questo caso nella tabella delle corrispondenze oltre all'indirizzo base del segmento, ci sta anche la sua lunghezza