

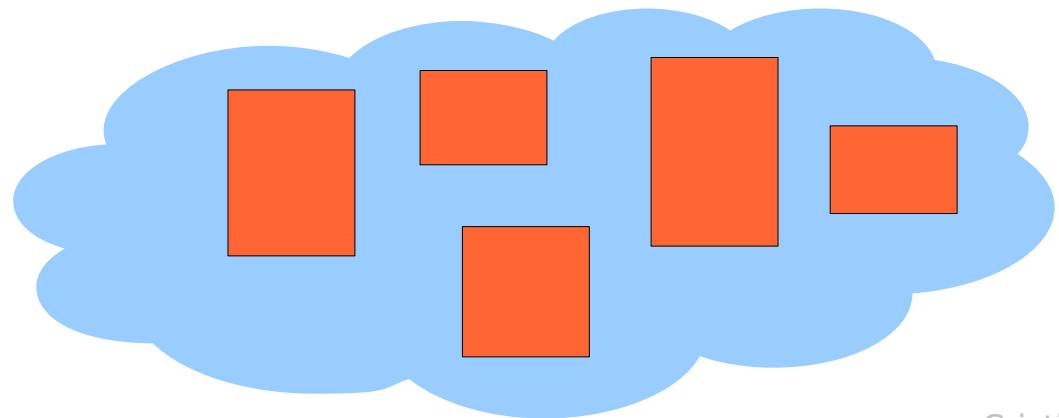


segmentazione

capitolo 8 del libro (VII ed.), da 8.6

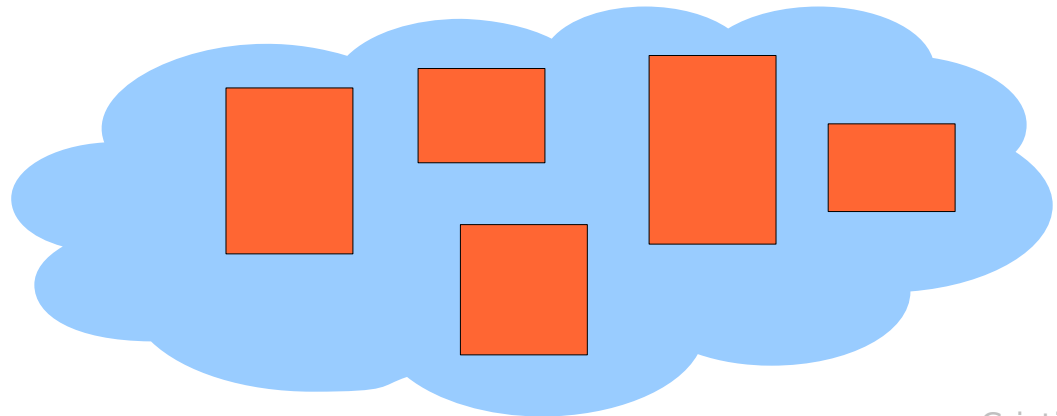
Introduzione

- La paginazione struttura la memoria in un insieme di elementi, il cui contenuto può essere indifferentemente costituito da codice e/o dati
- Questa organizzazione è molto diversa dal modo in cui i *programmatori* vedono i programmi, cioè come strutturate in parti con un preciso *valore funzionale* (il codice, le librerie, lo stack, l'heap)



Introduzione

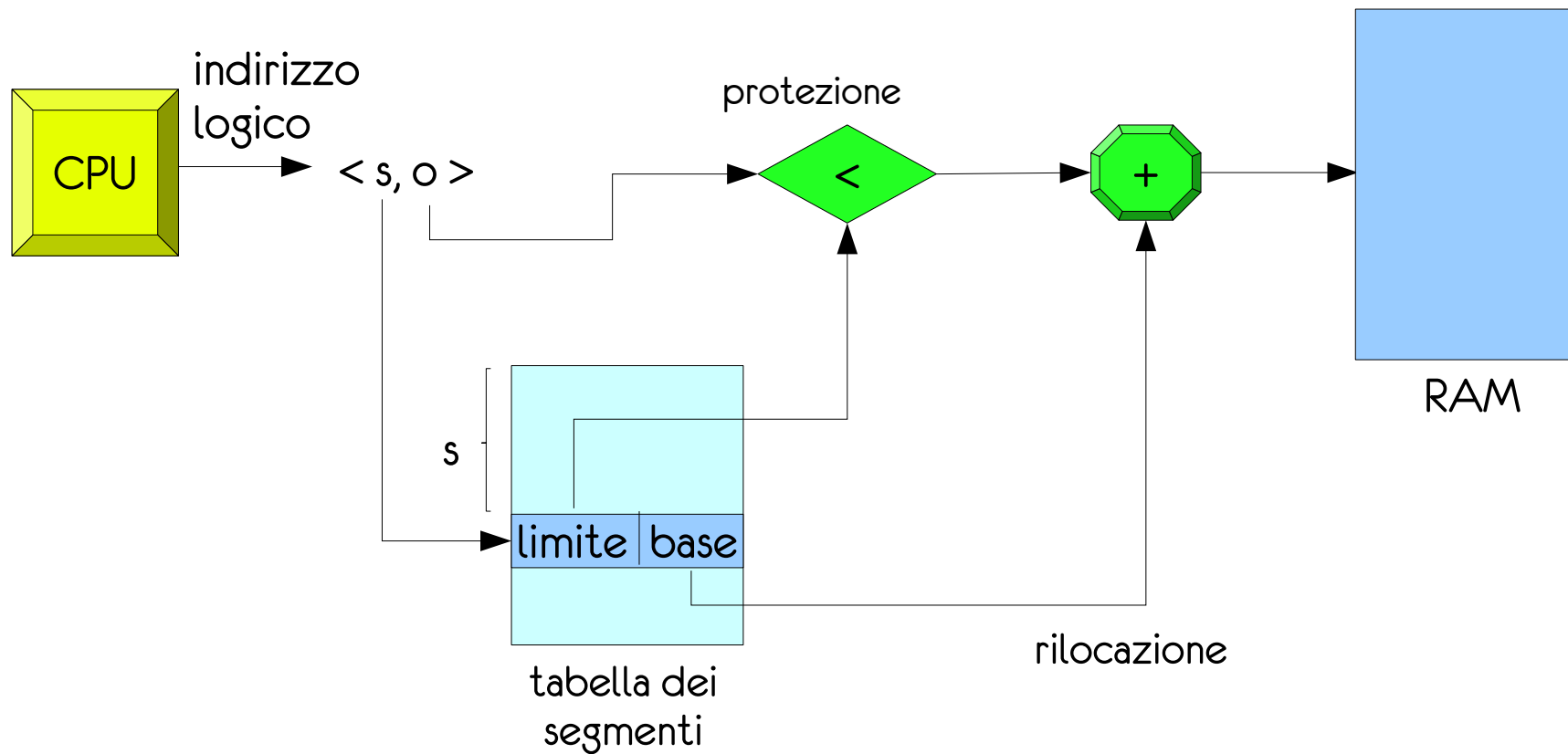
- La segmentazione modella la memoria secondo uno schema più vicino al modo in cui il programmatore vede il programma
- Ogni processo ha una porzione di RAM organizzata come un insieme di segmenti di memoria, di dimensione variabile



Segmenti

- Per adottare questo approccio è necessario che il programma risulti già organizzato in segmenti. Tale strutturazione viene effettuata dal compilatore
- Es. compilatore C può creare questi segmenti per un programma:
 - codice
 - vrb globali
 - stack per ciascun thread
 - heap
 - libreria standard del C
 - le altre librerie avranno assegnati segmenti in un tempo successivo
- Indirizzo logico: **< id_di_segmento, offset >**

Architettura



s = indice della entry relativa al segmento

ogni segmento è caratterizzato da due valori: **indirizzo di base** e **indirizzo limite** perché la loro dimensione è varia

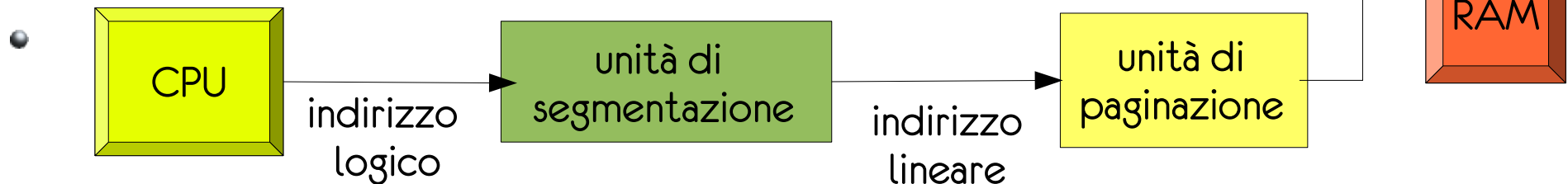
Commenti

- La segmentazione è spesso più efficiente della paginazione
- I segmenti possono avere dimensione estremamente varia
- Quando lo scheduler a medio o lungo termine carica un processo, deve trovare uno spazio adeguato alle dimensioni dei suoi segmenti.
- I problemi sono simili a quelli dello schema a partizioni multiple, anche se la segmentazione è un meccanismo molto più flessibile
 - occorre gestire la memoria libera in modo dinamico
 - si ha frammentazione esterna
 - possibile attuare tecniche di compattazione per ridurre la frammentazione

paginazione + segmentazione

- Alcune architetture sfruttano una tecnica che si basa sull'unione di paginazione e segmentazione

- **Esempio: Pentium Intel**



Ogni segmento è strutturato in pagine

dimensione max segmento 4GB
numero max di segmenti per processo 16K

Il processore ha 6 registri di segmento che permettono a un processo di fare riferimento a 6 segmenti contemporaneamente

8K riservati ∈ tabella locale dei descrittori

8K condivisi ∈ tabella globale dei descrittori

Intel Pentium

indirizzo logico

< segmento, g, o >

id di segmento

offset: 32 bit

segmento globale/locale



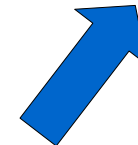
indirizzo lineare

ottenuto tramite base e limite del segmento
+ offset: non è ancora un indirizzo fisico

Si applica infine uno schema
di **paginazione a due livelli**

Indirizzo lineare così scomposto:

< p1, p2, offset >



Conclusioni

- Esistono diversi modelli per la gestione della RAM
- La scelta dipende fortemente dall'HW a disposizione
- Ogni HW è progettato in modo tale da supportare uno (o più) modelli specifici
- Nel valutare un modello/HW occorre tenere presente:
 - se consente la rilocalizzazione
 - se consente lo swapping
 - se consente la condivisione
 - se attua meccanismi di protezione

supporti alla
multi-programmazione