Gestione della memoria principale

Gestione della memoria

Il S.O deve:

- tener traccia delle aree di memoria principale **occupate** da ciascun processo, e delle aree **libere**
- stabilire in quali aree libere caricare i nuovi processi (**tecnica di allocazione** della memoria)
- impedire che un processo possa accedere liberamente all'area di memoria di un altro processo (**protezione** della memoria)
- decidere cosa fare nel caso in cui i processi avviati richiedano un quantità di memoria maggiore di quella fisicamente disponibile (gestione della **memoria virtuale**)

Indirizzi fisici e logici

Prima di discutere le possibili strategie di gestione della memoria, occorre chiarire la differenza tra indirizzi fisici e indirizzi logici.

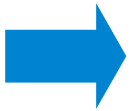
La memoria principale (RAM) è costituita da una sequenza di N locazioni di memoria (o celle) di 1 byte, ognuna delle quali è identificata da un numero intero compreso tra 0 e N-1. Chiamiamo questo numero **indirizzo fisico** della locazione.

Esempio: se un computer ha 1GB di RAM, significa che la RAM contiene 2^{30} locazioni da 1 byte. Il primo byte ha indirizzo fisico 0, e l'ultimo byte avrà indirizzo fisico 2^{30}-1.

Indirizzi fisici e logici

- Tutti i programmi in linguaggio macchina contengono istruzioni che coinvolgono (cioè "fanno riferimento a") indirizzi di memoria
- Esempio: istruzioni LOAD e STORE

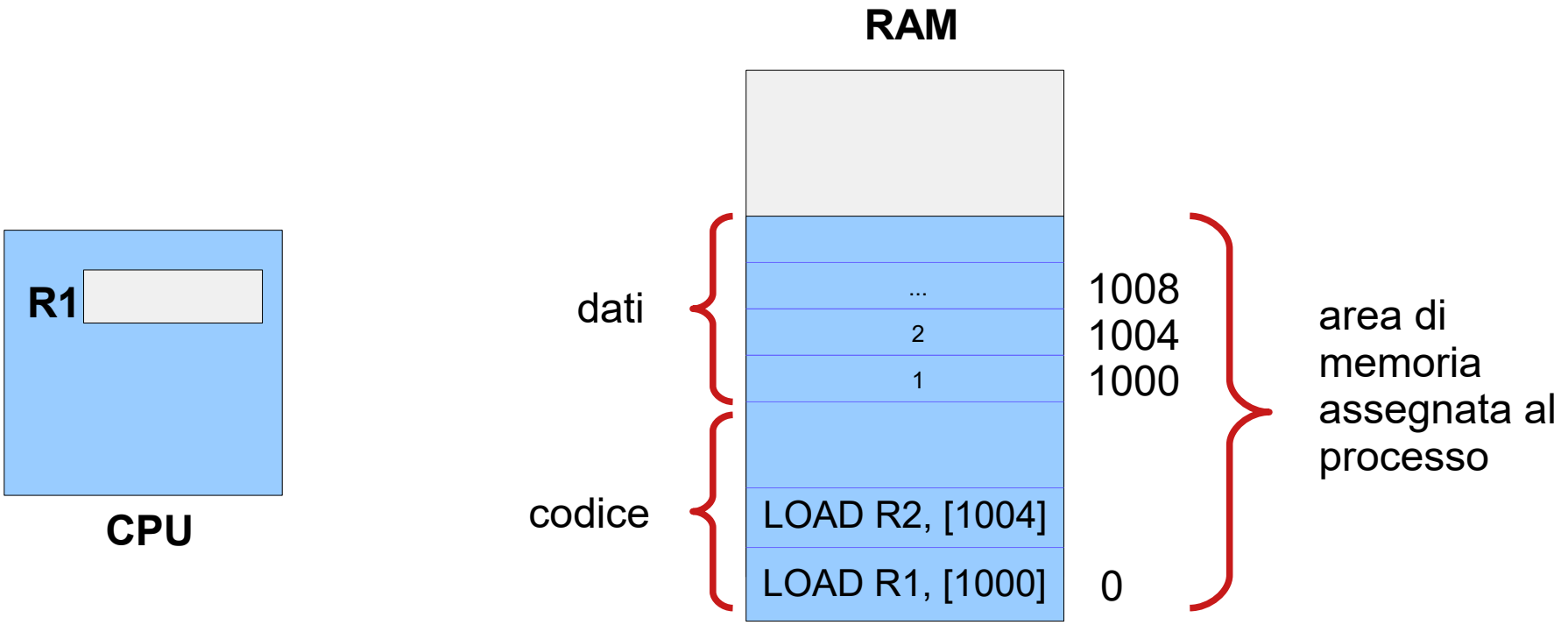
```
int a=1,b=2,c;
a = b + c;
```



```
LOAD R1, [1000]
LOAD R2, [1004]
ADD R2, R1
STORE [1008], R2
```

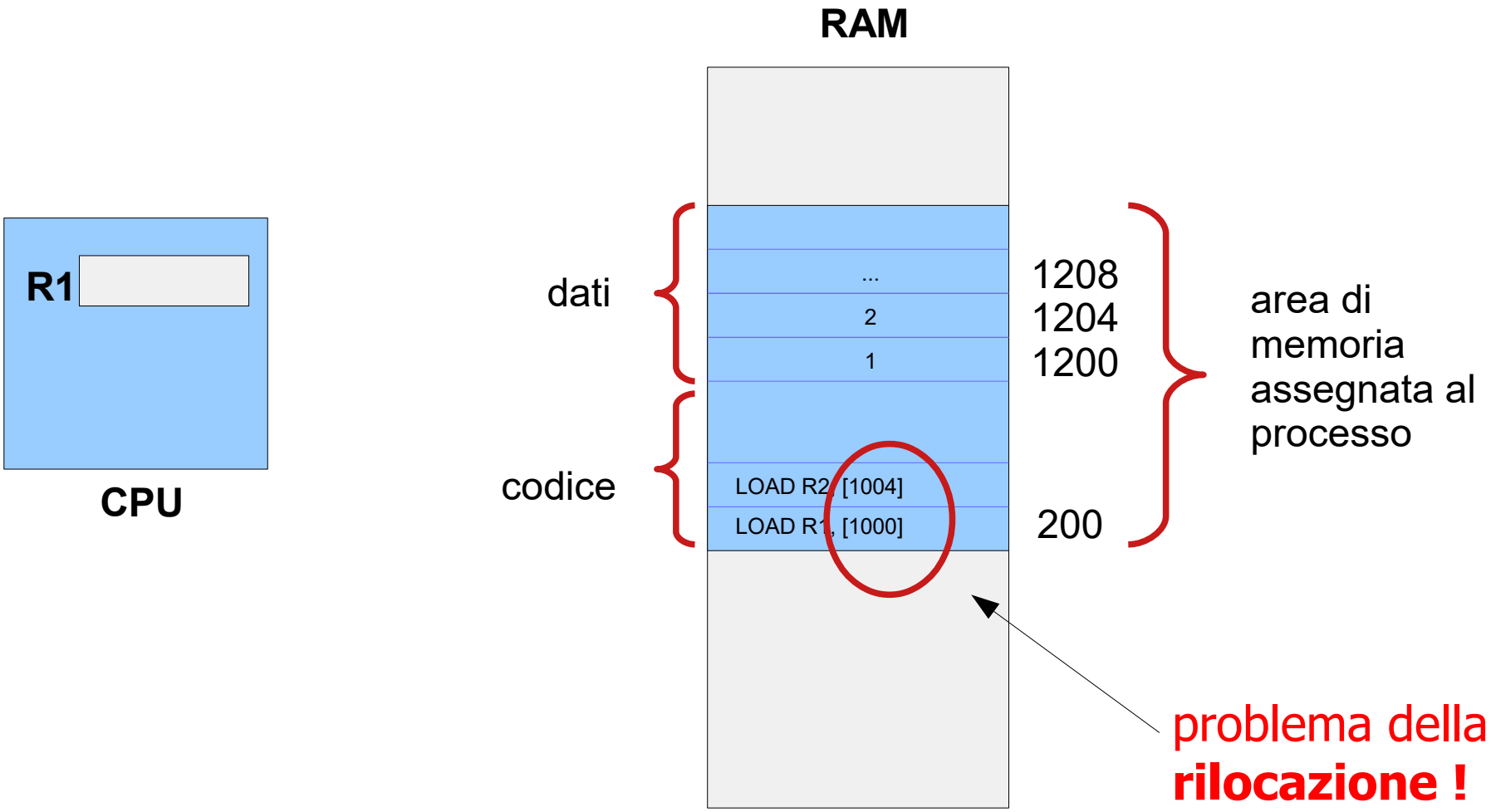
Indirizzi fisici e logici

Se gli indirizzi contenuti nel programma eseguibile fossero interpretati come indirizzi fisici, il programma funzionerebbe solo se caricato in memoria a partire dall'indirizzo 0



Indirizzi fisici e logici

Ma cosa succede se il programma viene caricato in un'area di memoria che ha un indirizzo fisico iniziale diverso da 0 ?



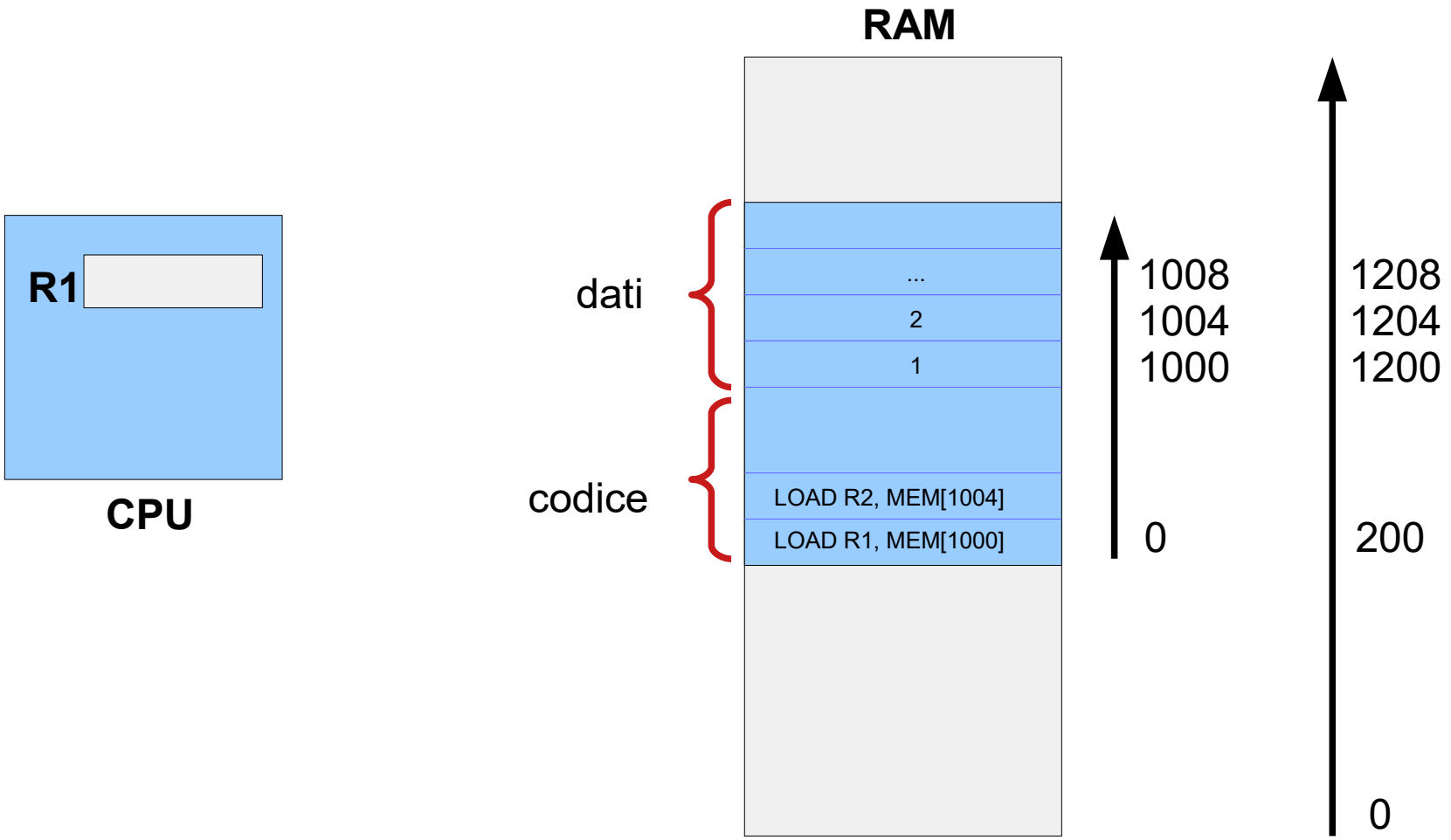
Indirizzi fisici e logici

Negli attuali sistemi operativi per PC, gli indirizzi di memoria contenuti nelle istruzioni macchina non possono essere indirizzi fisici perchè:

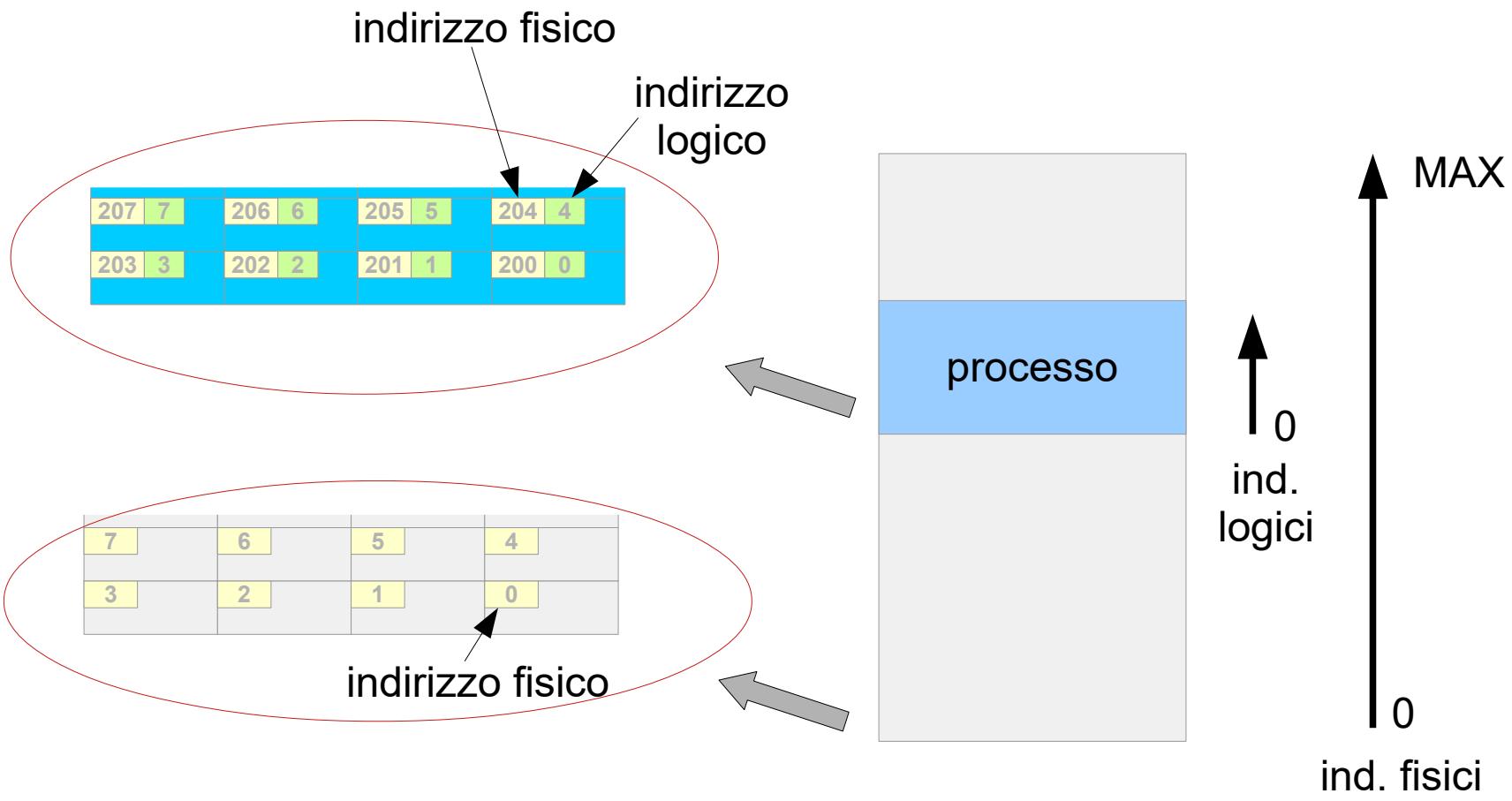
- gli attuali sistemi operativi sono in grado di eseguire "contemporaneamente" più programmi e, per realizzare ciò in modo efficiente, più programmi devono essere caricati contemporaneamente in memoria principale.
- il compilatore non può sapere in quale zona della memoria fisica il programma sarà effettivamente caricato (ciò dipende da quali zone di memoria saranno libere nel momento in cui l'utente deciderà di eseguite il programma) e quindi non può generare indirizzi fisici.

Indirizzi logici

In questi sistemi, il compilatore può solamente generare gli indirizzi *come se* il programma venisse sempre caricato in un'area di memoria consecutiva che inizia all'indirizzo zero. Gli indirizzi così generati sono detti indirizzi logici e possoono essere considerati come indirizzi relativi all'inizio del programma, anzichè all'inizio della memoria



Riassumendo



Indirizzi fisici e logici in C

I puntatori in C contengono indirizzi fisici o logici ?

- Gli indirizzi di memoria utilizzabili nei programmi C/C++ di livello utente sui sitemi operativi Windows o Linux sono in realtà sempre indirizzi logici.
- In altre parole, su questi sistemi il programmatore di software applicativo non ha accesso agli indirizzi fisici, e un processo non puà sapere in quale parte della memoria RAM è stato effettivamente caricato.
- Su questi sistemi, l'operatore & del C restituisce l'indirizzo logico della variabile, e quindi i puntatori contengono indirizzi logici.
- ATTENZIONE: in alcuni casi (programmazione di microcontrollori, sw di livello kernel, vecchi sistemi operativi, ..) i programmatori possono referenziare in modo assoluto la memoria (cioè possono utilizzare gli indirizzi fisici). In questi casi i puntatori contengono indirizzi fisici.

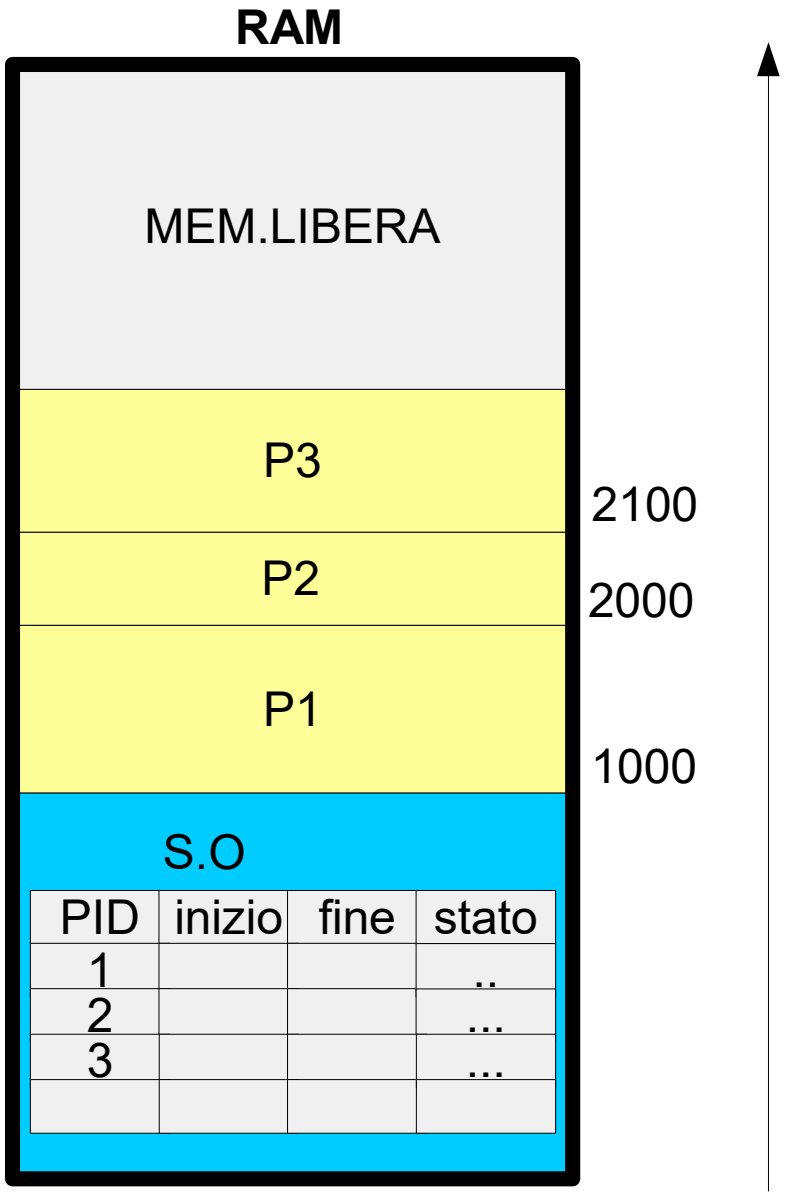
Teniche di gestione della memoria

Possibibili tecniche di gestione della memoria:

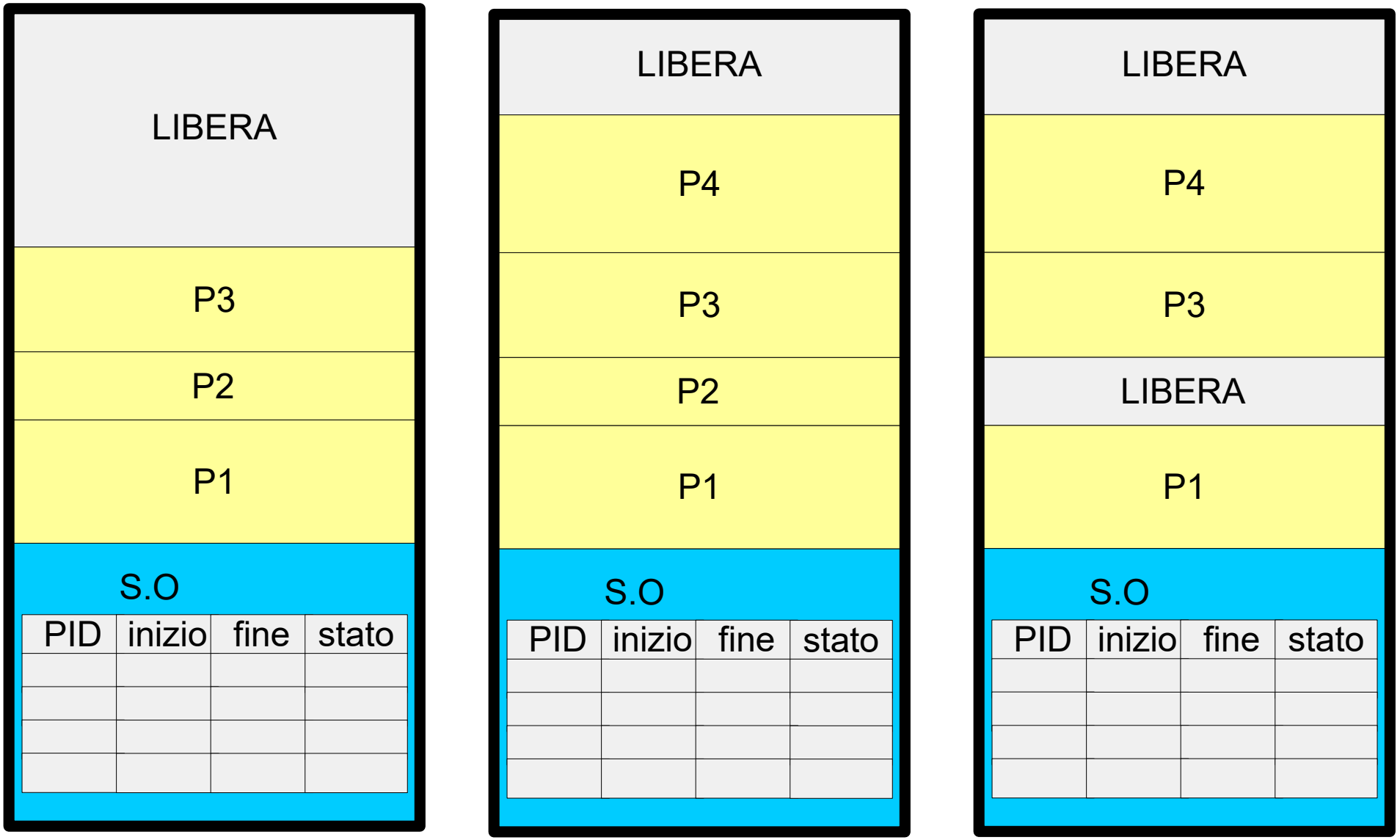
- Allocazione contigua
- Paginazione
- Segmentazione
- Segmentazione + Paginazione

Allocazione contigua della memoria

- è il metodo più semplice: ogni processo occupa un'unica area di memoria, la cui dimensione è pari alla quantità di memoria ad esso necessaria (codice + dati).
- il processo è caricato in locazioni di memoria fisica consecutive, a partire da un certo indirizzo fisico minimo, fino ad un indirizzo fisico massimo.
- svantaggio: esiste il problema della frammentazione della memoria

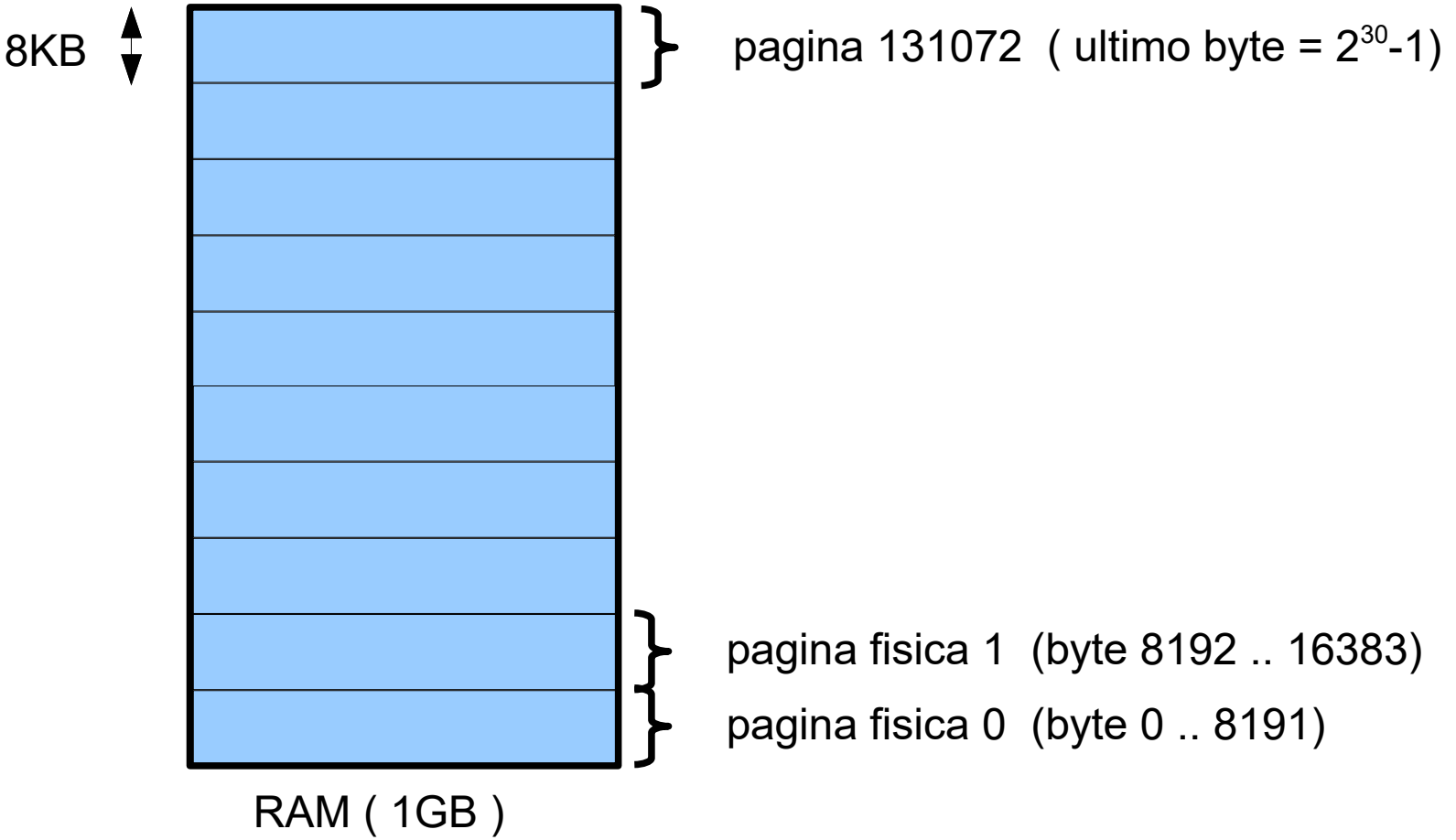


Frammentazione



Paginazione

La paginazione è una tecnica di gestione della memoria che prevede la suddivisione della memoria principale in **blocchi di dimensione fissa chiamati pagine fisiche**. Dimensione tipica delle pagine: 4KB – 8KB (stabilita dal sitema operativo)



Paginazione

- Con la paginazione, un processo può occupare un qualsiasi insieme di pagine fisiche, non necessariamente consecutive
- Per realizzare ciò, anche l'insieme degli indirizzi logici di un programma è suddiviso in blocchi di dimensione fissa, detti **pagine logiche** (di dimensione uguale alle pagine fisiche)
- Esempio: con pagine di 4KB, i primi 4096 byte del programma (ovvero i byte con indirizzo logico compreso tra 0 e 4095) costituiscono la "pagina logica 0" , i successivi 4096 byte costituiscono la "pagina logica 1", e così via.
- Quando un programma deve essere caricato in memoria per essere eseguito, **ogni pagina logica del programma può essere posta in una qualsiasi pagina fisica**. Si elimina così il problema della frammentazione presente nell'allocazione contigua ("problema della frammentazione esterna").

Paginazione: esempio

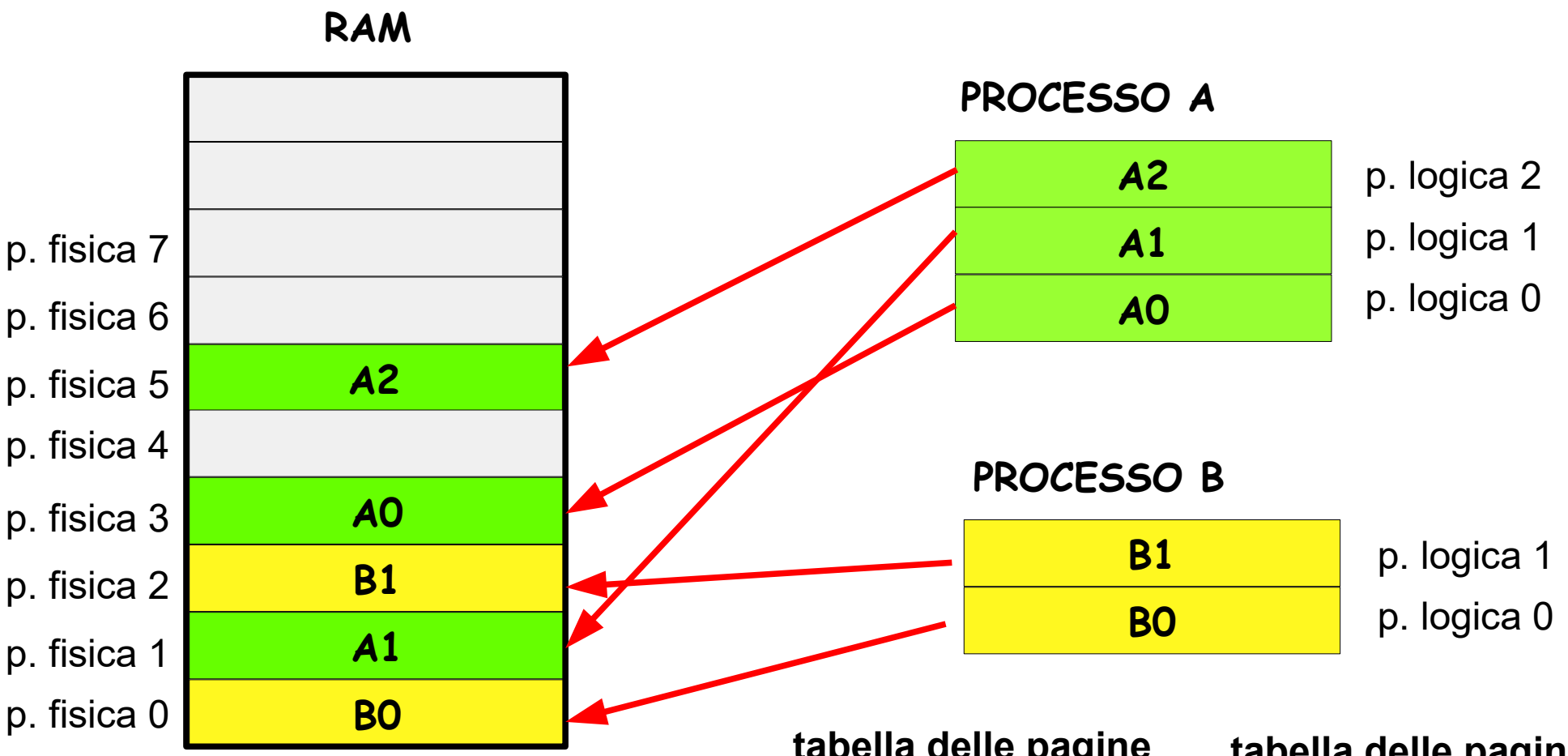


tabella delle pagine del processo A

Nr. pag. logica	Nr. pag. fisica
0	3
1	1
2	5

tabella delle pagine del processo B

Nr. pag. logica	Nr. pag. fisica
0	0
1	2

Errore di pagina (page fault)

Se una nuova pagina logica deve essere caricata in RAM e non vi sono più pagine fisiche libere (o il processo sta già occupando il massimo numero di pagine ad esso consentito) si dice che si è verificato un "page-fault" (errore di pagina). Quando si verifica un page-fault, il sistema operativo deve:

1) individuare una pagina fisica da sostituire

2) ricopiare il contenuto di questa pagina su disco (in una zona chiamata area di swap)

3) copiare il contenuto della nuova pagina logica nella pagina fisica individuata al punto 1

Politiche di sostituzione delle pagine

Per la scelta della pagina fisica da sostituire, il S.O può utilizzare diverse strategie:

- FIFO (First In First Out)
- LRU (Least Recently Used)
- LFU (Least Frequently Used)

Nota: i sistemi operativi attuali utilizzano tecniche più complesse e la politica effettivamente utilizzata dipende anche dalle caratteristiche hardware disponibili.

Esercizio

Considera l'attuale tabella delle pagine del processo A:

Nr. pag. logica	Nr. pag. fisica	...	Istante caricamento (ms)	Istante ultimo accesso (ms)	Numero accessi
0	5		1000	3300	300
1	3		2000	2100	200
2	1		3000	3100	100

In un istante successivo, il processo A richiede l'accesso alla propria pagina logica 3, non ancora presente in RAM.

Supponendo che non vi siano pagine fisiche libere in RAM, quale pagina fisica, fra quelle del processo A, sarebbe rimossa dalla RAM se il S.O utilizza una politica di sostituzione FIFO?

Rispondi alla stessa domanda supponendo di utilizzare la politica LRU e poi LFU.

Soluzione

Nr. pag. logica	Nr. pag. fisica	...	Istante caricamento (ms)	Istante ultimo accesso (ms)	Numero accessi
0	5		1000	3300	300
1	3		2000	2100	200
2	1		3000	3100	100