

Programmazione e Laboratorio di Programmazione

Lezione IV

Il modello di memoria

Il modello di memoria

2834		
2835		
2836		
2837		
2838		
2839		
2840		
2841		
2842	10100000	X
2843		
2844		
2855		

- un insieme di **locazioni**
- ogni locazione è univocamente identificata da un valore numerico detto **indirizzo**
- ogni locazione memorizza una **sequenza di 8 bit**
- ad ogni locazione è associato un **flag** che indica se la locazione è correntemente in uso o è invece disponibile

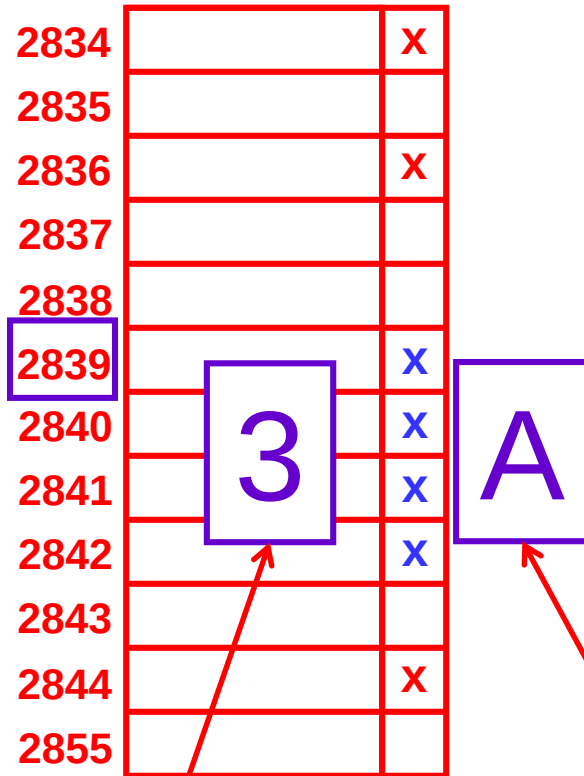


In uso



Disponibile

Variabili



- una **variabile** è un insieme di locazioni contigue in memoria
- ogni variabile è identificata da una stringa alfa-numerica detta **nome** della variabile
- l'**indirizzo** di una variabile è l'indirizzo della prima di tali locazioni
- ogni variabile memorizza un **valore**
- il **tipo** della variabile è il tipo dei valori memorizzabili
- il **numero delle locazioni** "allocate" per una variabile è funzione del tipo della variabile

intero

4 locazioni x un valore intero

Stato della memoria

- **Molto informalmente:**

è una foto del contenuto della memoria ad un certo istante

- **Molto meno informalmente:**

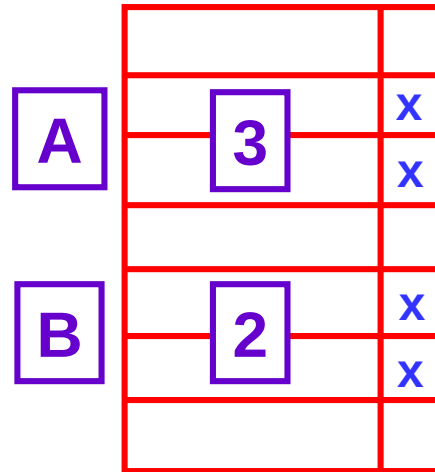
è determinato dall'insieme delle quadruple

$(\text{nome}_{\text{var}}, \text{tipo}_{\text{var}}, \text{valore}_{\text{var}}, \text{indirizzo}_{\text{var}})$

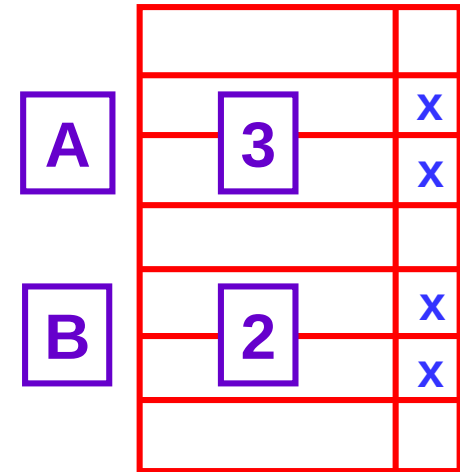
Stato della memoria

Stato₁ =
Stato₂?

SI



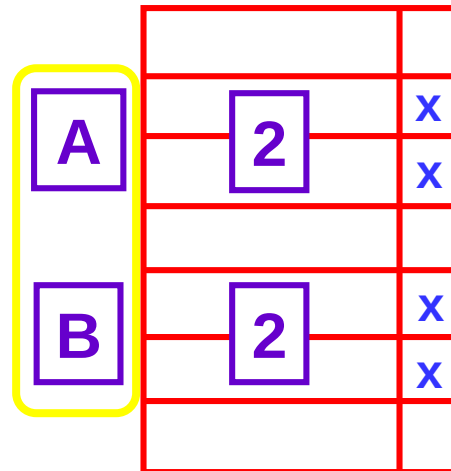
Stato₁



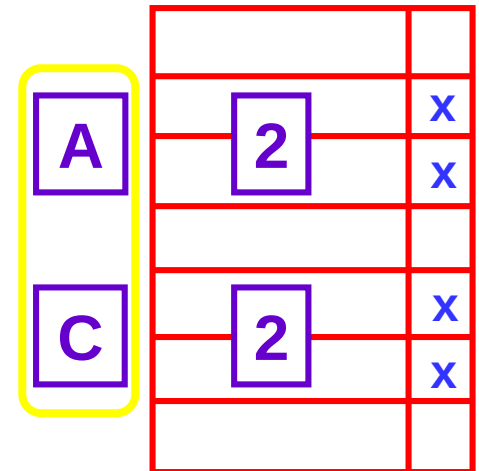
Stato₂

Stato₁ =
Stato₂?

NO



Stato₁

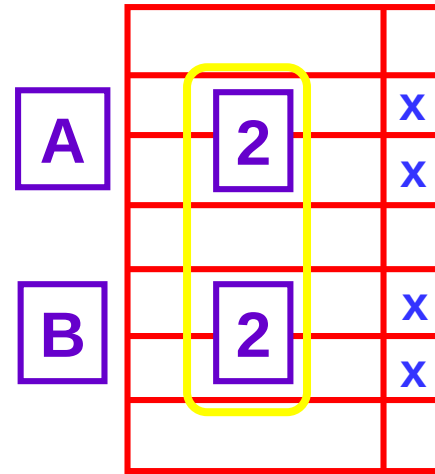


Stato₂

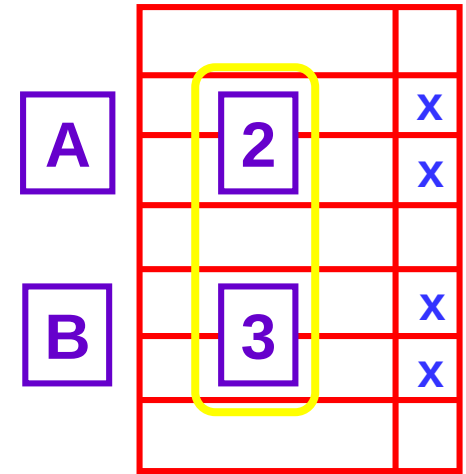
Stato della memoria

$\text{Stato}_1 =$
 $\text{Stato}_2?$

NO



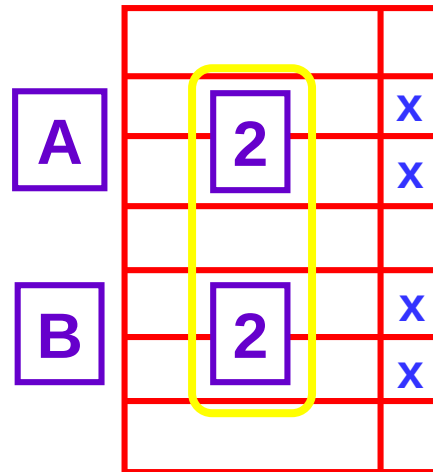
Stato_1



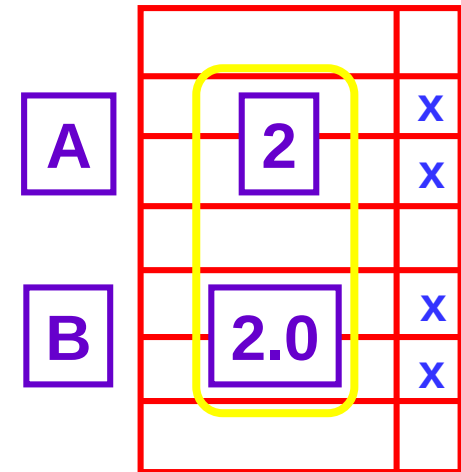
Stato_2

$\text{Stato}_1 =$
 $\text{Stato}_2?$

NO



Stato_1

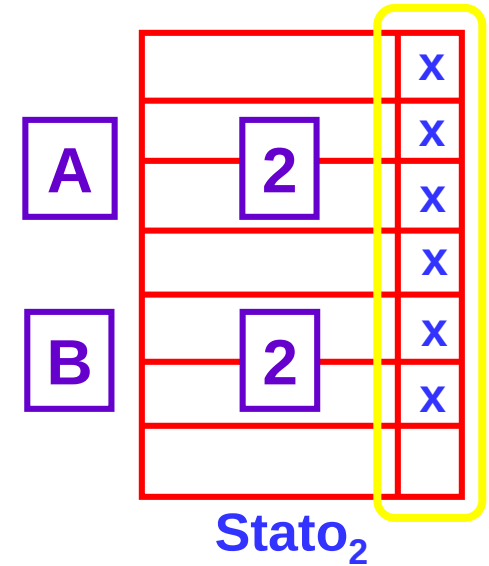
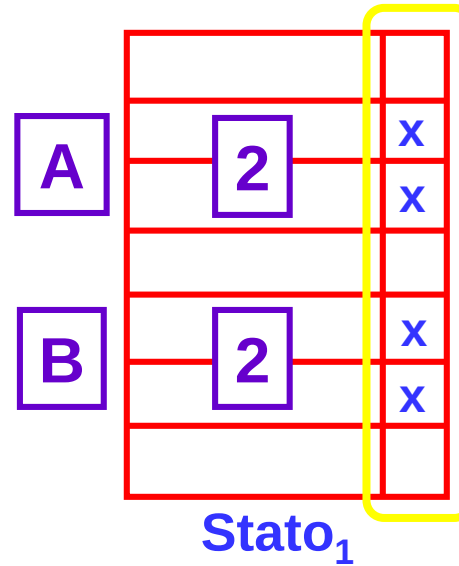


Stato_2

Stato della memoria

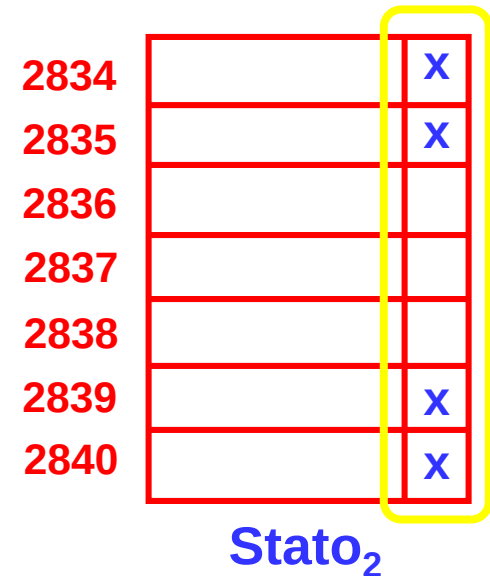
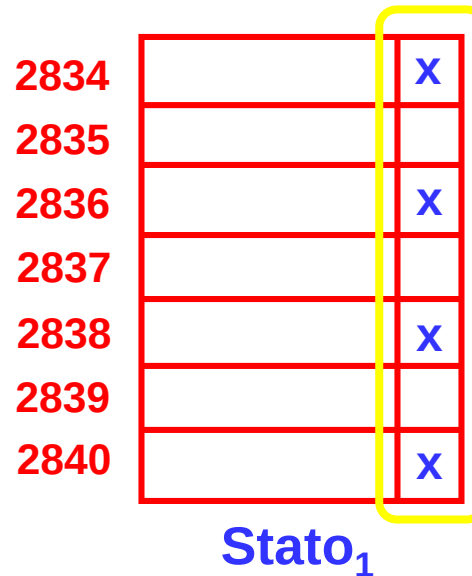
Stato₁ =
Stato₂?

NO



Stato₁ =
Stato₂?

NO



Definizione di una variabile

- **Definizione di una variabile:**

tipo_variabile nome_variabile

- **Modifiche allo stato della memoria:**

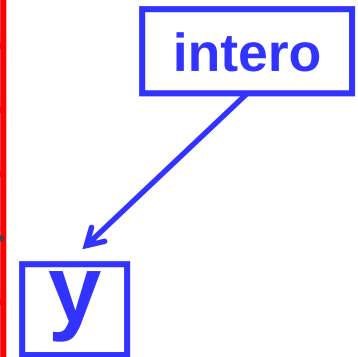
1. si riserva il numero di locazioni contigue in memoria necessario a memorizzare un valore di tipo **tipo_variabile**
2. si associa il nome **nome_variabile** alle locazioni riservate
3. si associa alla variabile il tipo **tipo_variabile**

Definizione di una variabile

- **Esempio:**

int y;

2834		x
2835		
2836		x
2837		
2838		
2839		x
2840		x
2841		x
2842		x
2843		
2844		x
2855		



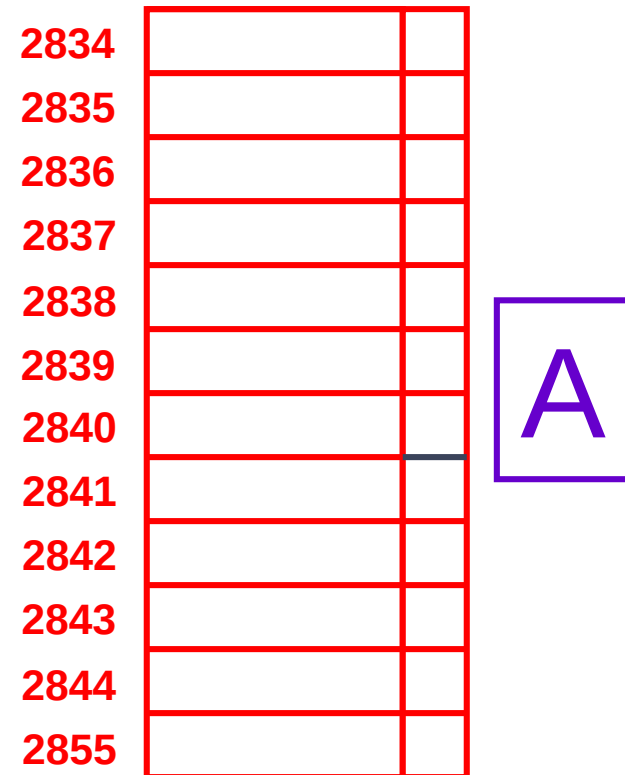
Rilascio di memoria allocata

- **Rilascio della memoria:**

avviene rendendo di nuovo disponibili
le celle allocate per la variabile

- **Esempio:**

```
{  
    {  
        → int A;  
        ....  
        → }  
    }  
}
```



Assegnamento di valori a variabili

- **Assegnamento:**

nome_variabile = espressione

- **Modifiche allo stato della memoria:**

1. si valuta il valore di **espressione**
2. si sostituisce tale valore al valore memorizzato da **nome_variabile**

Attenzione:

il tipo di **nome_variabile** e il tipo di **espressione** “devono” essere gli stessi

Accesso alle variabili tramite nome

- Esempio:

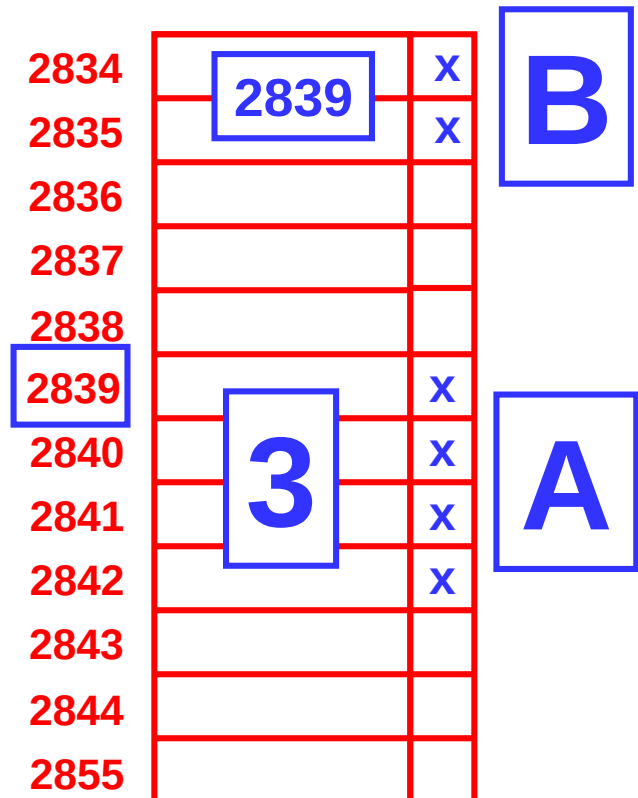
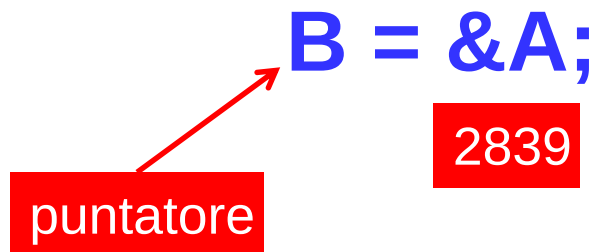
-4 **3**
B = A+C;

2834		x	
2835	-1	x	B
2836		x	
2837		x	
2838		x	
2839	3	x	C
2840		x	
2841		x	
2842		x	
2843	-4	x	A
2844		x	
2845		x	
2846		x	

Accesso all'indirizzo di una variabile

- **L'operatore &:**
`&nome_variabile`
- **Valore restituito:**

l'indirizzo di `nome_variabile`



Definizione di un puntatore

- **Definizione di un puntatore:**

tipo_variabile_riferita* nome_puntatore

- **Modifiche allo stato della memoria:**

1. si riserva il numero di locazioni contigue in memoria necessario a memorizzare un indirizzo
2. si associa il nome **nome_variabile** alle locazioni riservate
3. si associa alla variabile il tipo "indirizzo di una variabile di tipo **tipo_variabile_riferita**"

Accesso alle variabili tramite indirizzo

- **L'operatore *:**

***nome_puntatore**

- **Effetto:**

equivale al nome della
variabile il cui indirizzo
è in **nome_puntatore**

- **Esempio:**

3
A
A=*B+1;

2834	2839	x	B
2835		x	
2836			
2837			
2838			A
2839		x	
2840	4	x	
2841		x	
2842		x	
2843			
2844			
2855			

Accesso alle variabili tramite indirizzo

- Esempio:

C **4** **3**
***B = A+C;**

