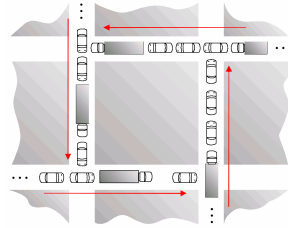


Esercizi

Stallo dei processi



Per riscaldarsi: esercizio 0

Provare che il sistema è in stallo
Quali sono le risorse?
Chi reclama le risorse?

soluzione

Risorse: incroci che devono essere attraversati

Processi: file di auto che devono attraversare gli incroci

Condizioni per lo stallo:

1: **mutua esclusione** (gli incroci non possono essere condivisi)

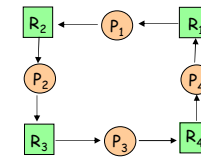
2: **possesso e attesa** (ogni fila di auto possiede un incrocio e aspetta che se ne liberi uno)

3: **non prelazionabilità** (le auto non possono essere spostate)

4: **attesa circolare**

■ risorse

○ processi



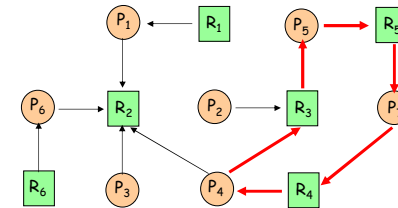
Esercizio 1

Un sistema è composto da **7 processi**, P1, ..., P7, e da **6 risorse** condivise, R1, ..., R6, ciascuna di tipo diverso, non prelazionabili e ad accesso mutuamente esclusivo. La situazione del sistema è la seguente:

- P1 **occupa** R1 e **richiede** R2;
- P2 **non occupa** risorse e **richiede** R3;
- P3 **non occupa** risorse e **richiede** R2;
- P4 **occupa** R4 e **richiede** sia R2 sia R3;
- P5 **occupa** R3 e **richiede** R5;
- P6 **occupa** R6 e **richiede** R2;
- P7 **occupa** R5 e **richiede** R4.

Si determini, utilizzando il grafo di allocazione delle risorse, **se il sistema è in deadlock** e, in caso affermativo, quali sono i **processi e le risorse coinvolti**.

soluzione



$P4 \rightarrow R3 \rightarrow P5 \rightarrow R5 \rightarrow P7 \rightarrow R4 \rightarrow P4$ sono in deadlock.

Esercizio 2

Dati 4 processi, e 3 tipi di risorse: A (5 istanze), B (8 istanze), e C (16 istanze).

	Assegnate			Massimo			Rtot		
	A	B	C	A	B	C	A	B	C
P1	0	1	4	4	1	4	5	8	16
P2	2	0	1	3	1	4			
P3	1	2	1	5	7	13			
P4	1	0	3	1	1	6			

Si determini:

- se il sistema è in uno stato sicuro;
- se l'assegnazione di 1 istanza della risorsa A a P1 garantisce il mantenimento dello stato sicuro;
- se l'assegnazione di 6 istanze di C a P3 garantisce il mantenimento dello stato sicuro

Soluzione a)

PASSO 1

	Assegnate			Massimo			Disponibili			Necessita'		
	A	B	C	A	B	C	A	B	C	A	B	C
P1	0	1	4	4	1	4	1	5	7	4	0	0
P2	2	0	1	3	1	4				1	1	3
P3	1	2	1	5	7	13				4	5	12
P4	1	0	3	1	1	6				0	1	3

Disponibili = Rtot - Assegnate
Necessita' = Massimo - Assegnate

- e' possibile soddisfare le richieste di P₂
- Alla fine dell'esecuzione P₂ rilascia le risorse → Disponibili = (3 5 8)
- Sequenza sicura < P₂ >

Soluzione a)

PASSO 2

	Assegnate			Massimo			Disponibili			Necessita'		
	A	B	C	A	B	C	A	B	C	A	B	C
P1	0	1	4	4	1	4	3	5	8	4	0	0
P2	2	0	1	3	1	4				0	0	0 terminato
P3	1	2	1	5	7	13				4	5	12
P4	1	0	3	1	1	6				0	1	3

- e' possibile soddisfare le richieste di P₄
- Alla fine dell'esecuzione P₄ rilascia le risorse → Disponibili = (4 5 11)
- Sequenza sicura < P₂, P₄ >

Soluzione a)

PASSO 3

	Assegnate			Massimo			Disponibili			Necessita'		
	A	B	C	A	B	C	A	B	C	A	B	C
P1	0	1	4	4	1	4	4	5	11	4	0	0
P2	2	0	1	3	1	4				0	0	0 terminato
P3	1	2	1	5	7	13				4	5	12
P4	1	0	3	1	1	6				0	0	0 terminato

- e' possibile soddisfare le richieste di P₁
- Alla fine dell'esecuzione P₁ rilascia le risorse → Disponibili = (4 6 15)
- Sequenza sicura < P₂, P₄, P₁ >

Soluzione a)

PASSO 4

	<i>Assegnate</i>	<i>Massimo</i>	<i>Disponibili</i>	<i>Necessita'</i>	
	A B C	A B C	A B C	A B C	
P1	0 1 4	4 1 4	4 6 15	0 0 0	terminato
P2	2 0 1	3 1 4		0 0 0	terminato
P3	1 2 1	5 7 13		4 5 12	
P4	1 0 3	1 1 6		0 1 3	terminato

- e' possibile soddisfare le richieste di P₃
- Alla fine dell'esecuzione P₃ rilascia le risorse → Disponibili = (5 8 16)
- Sequenza sicura < P₂, P₄, P₁, P₃ >

IL SISTEMA E' IN STATO SICURO

Soluzione b)

PASSO 1

	<i>Assegnate</i>	<i>Massimo</i>	<i>Disponibili</i>	<i>Necessita'</i>
	A B C	A B C	A B C	A B C
P1	1 1 4	4 1 4	0 5 7	3 0 0
P2	2 0 1	3 1 4		1 1 3
P3	1 2 1	5 7 13		4 5 12
P4	1 0 3	1 1 6		0 1 3

Disponibili = R_{tot} - Assegnate
 Necessita' = Massimo - Assegnate

- e' possibile soddisfare le richieste di P₄
- Alla fine dell'esecuzione P₄ rilascia le risorse → Disponibili = (1 5 10)
- Sequenza sicura < P₄ >

Soluzione b)

PASSO 2

	<i>Assegnate</i>	<i>Massimo</i>	<i>Disponibili</i>	<i>Necessita'</i>
	A B C	A B C	A B C	A B C
P1	1 1 4	4 1 4	1 5 10	3 0 0
P2	2 0 1	3 1 4		1 1 3
P3	1 2 1	5 7 13		4 5 12
P4	1 0 3	1 1 6		0 0 0

- e' possibile soddisfare le richieste di P₂
- Alla fine dell'esecuzione P₂ rilascia le risorse → Disponibili = (3 5 11)
- Sequenza sicura < P₄, P₂ >

Soluzione b)

PASSO 3

	<i>Assegnate</i>	<i>Massimo</i>	<i>Disponibili</i>	<i>Necessita'</i>
	A B C	A B C	A B C	A B C
P1	1 1 4	4 1 4	3 5 11	3 0 0
P2	2 0 1	3 1 4		0 0 0
P3	1 2 1	5 7 13		4 5 12
P4	1 0 3	1 1 6		0 0 0

- e' possibile soddisfare le richieste di P₁
- Alla fine dell'esecuzione P₁ rilascia le risorse → Disponibili = (4 6 15)
- Sequenza sicura < P₄, P₂, P₁ >

Soluzione b)

PASSO 4

	<i>Assegnate</i>	<i>Massimo</i>	<i>Disponibili</i>	<i>Necessita'</i>	
	A B C	A B C	A B C	A B C	
P1	1 1 4	4 1 4	4 6 15	0 0 0	terminato
P2	2 0 1	3 1 4		0 0 0	terminato
P3	1 2 1	5 7 13		4 5 12	
P4	1 0 3	1 1 6		0 0 0	terminato

- e' possibile soddisfare le richieste di P₃
- Alla fine dell'esecuzione P₃ rilascia le risorse → Disponibili = (5 8 16)
- Sequenza sicura < P₄, P₂, P₁, P₃ >

L'aggiunta di una risorse di tipo A a P1 produce uno stato sicuro

Soluzione c)

PASSO 1

	<i>Assegnate</i>	<i>Massimo</i>	<i>Disponibili</i>	<i>Necessita'</i>
	A B C	A B C	A B C	A B C
P1	1 1 4	4 1 4	2 5 1	4 0 0
P2	2 0 1	3 1 4		1 1 3
P3	1 2 1	5 7 13		4 5 6
P4	1 0 3	1 1 6		0 1 3

Disponibili =
Rtot - Assegnate

Necessita' =
Massimo - Assegnate

Non possono essere soddisfatte le richieste di nessun processo
Lo stato non e' sicuro