

• **Esercizio 1**

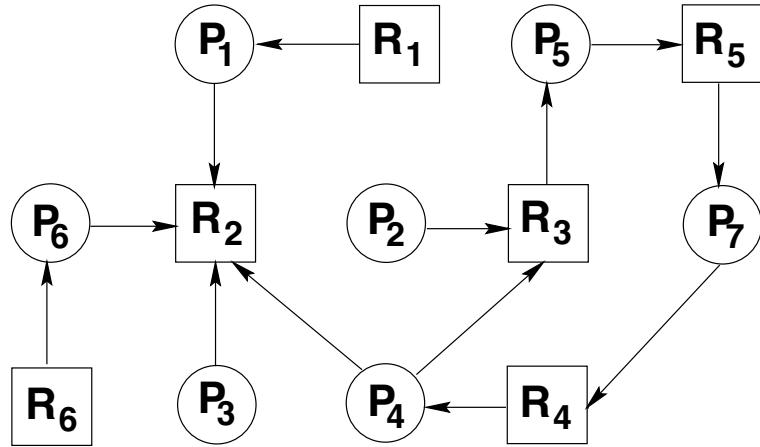
Un sistema è composto da 7 processi,  $P_1, \dots, P_7$ , e da 6 risorse condivise,  $R_1, \dots, R_6$ , ciascuna di tipo diverso. La situazione del sistema è la seguente:

- $P_1$  occupa  $R_1$  e richiede  $R_2$ ;
- $P_2$  non occupa risorse e richiede  $R_3$ ;
- $P_3$  non occupa risorse e richiede  $R_2$ ;
- $P_4$  occupa  $R_4$  e richiede sia  $R_2$  che  $R_3$ ;
- $P_5$  occupa  $R_3$  e richiede  $R_5$ ;
- $P_6$  occupa  $R_6$  e richiede  $R_2$ ;
- $P_7$  occupa  $R_5$  e richiede  $R_4$ .

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*

Il grafo di allocazione risorse è:



Il sistema è in deadlock. Il deadlock coinvolge i processi  $P_4$ ,  $P_5$  e  $P_7$  e le risorse  $R_3$ ,  $R_4$  e  $R_5$ .

- **Esercizio 2**

Si consideri un sistema costituito da 3 processi, in cui l'unico tipo di risorsa disponibile sia rappresentato da 12 unità a nastro. Utilizzando l'algoritmo del Banchiere di Dijkstra si stabilisca quando gli stati seguenti sono *sicuri* o *non sicuri*. Se uno stato è sicuro, si mostri secondo quale ordine i processi possano essere terminati.

– **Stato 1:**

Processo no.	Risorse allocate	Risorse max
$P_0$	1	4
$P_1$	4	6
$P_2$	5	8
Risorse disponibili	2	

– **Stato 2:**

Processo no.	Risorse allocate	Risorse max
$P_0$	8	10
$P_1$	2	5
$P_2$	1	3
Risorse disponibili	1	

Se uno stato è sicuro il sistema può comunque evolvere, a partire da quello stato, verso uno stato non sicuro. A partire dallo **Stato 1**, si supponga che a  $P_2$  sia assegnata una ulteriore istanza dell'unica risorsa disponibile. Com'è lo stato ottenuto?

*Soluzione*

Nello **Stato 1**, la successione di processi  $\langle P_1, P_0, P_2 \rangle$  è una successione sicura  $\Rightarrow$  il sistema è in stato sicuro. Viceversa, nello **Stato 2**, i processi  $P_0$  e  $P_2$ , che hanno le pretese minime, possono comunque richiedere 2 unità a nastri  $\Rightarrow$  lo stato del sistema è non sicuro. Ugualmente se, a partire dallo **Stato 1**, viene assegnata una ulteriore unità a nastri a  $P_2$  lo stato ottenuto è non sicuro.

• **Esercizio 3**

Si supponga di avere un sistema con 4 processi e 3 tipi di risorse disponibili. La matrice delle richieste da parte dei processi ha la struttura:

$$Max = \begin{bmatrix} 4 & 1 & 4 \\ 3 & 1 & 4 \\ 5 & 7 & 13 \\ 1 & 1 & 6 \end{bmatrix}$$

dove con  $Max(i, j)$  si indica il numero massimo di istanze della risorsa  $j$  richieste dal processo  $i$ . Il numero totale di risorse è espresso dal vettore:

$$R_{tot} = \begin{bmatrix} 5 & 8 & 16 \end{bmatrix}$$

mentre la matrice di allocazione è

$$Allocation = \begin{bmatrix} 0 & 1 & 4 \\ 2 & 0 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 3 \end{bmatrix}$$

$Allocation(i, j)$  rappresenta il numero di istanze della risorsa  $j$  che sono attualmente allocate al processo  $i$ . Si determini:

- a) se il sistema è attualmente in stato sicuro;
- b) se l'assegnazione di un'istanza della risorsa 1 al processo  $P_1$  garantisce il mantenimento dello stato sicuro;
- c) se l'assegnazione di 6 istanze della risorsa 3 al processo  $P_3$  garantisce il mantenimento dello stato sicuro.

*Soluzione*

- a) La matrice  $Need$  ed il vettore  $Available$  risultano in questo caso:

$$Need = \begin{bmatrix} 4 & 0 & 0 \\ 1 & 1 & 3 \\ 4 & 5 & 12 \\ 0 & 1 & 3 \end{bmatrix}, \quad Available = \begin{bmatrix} 1 & 5 & 7 \end{bmatrix}.$$

Conseguentemente la successione di processi  $\langle P_2, P_4, P_1, P_3 \rangle$  (che produce i vettori  $Available$   $[3,5,8]$ ,  $[4,5,11]$ ,  $[4,6,15]$ ,  $[5,8,16]$ ) è una successione sicura  $\Rightarrow$  il sistema è in stato sicuro.

- b) La matrice  $Need$  ed il vettore  $Available$  risultano in questo caso:

$$Need = \begin{bmatrix} 3 & 0 & 0 \\ 1 & 1 & 3 \\ 4 & 5 & 12 \\ 0 & 1 & 3 \end{bmatrix}, \quad Available = \begin{bmatrix} 0 & 5 & 7 \end{bmatrix}.$$

Conseguentemente la successione di processi  $\langle P_4, P_2, P_1, P_3 \rangle$  (che produce i vettori  $Available$   $[1,5,10]$ ,  $[3,5,11]$ ,  $[4,6,15]$ ,  $[5,8,16]$ ) è una successione sicura  $\Rightarrow$  il sistema è in stato sicuro.

- c) La matrice  $Need$  ed il vettore  $Available$  risultano in questo caso:

$$Need = \begin{bmatrix} 4 & 0 & 0 \\ 1 & 1 & 3 \\ 4 & 5 & 6 \\ 0 & 1 & 3 \end{bmatrix}, \quad Available = \begin{bmatrix} 1 & 5 & 1 \end{bmatrix}.$$

Non esiste nemmeno un processo le cui necessità residue possano essere soddisfatte dall'attuale vettore  $Available$   $\Rightarrow$  il sistema non è in stato sicuro.

- **Esercizio 4**

Si consideri una situazione in cui vi sono tre processi  $P_1$ ,  $P_2$  e  $P_3$  e quattro classi di risorse: 4 unità a nastro, 2 plotter, 3 stampanti, ed una unità CD-ROM. Si suppongano ancora disponibili per l'uso 2 unità a nastro ed un plotter. Considerando l'algoritmo del Banchiere di Dijkstra si stabilisca se lo stato seguente è *sicuro* o *non sicuro*:

$$Allocation = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \end{bmatrix}, \quad Need = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{bmatrix}.$$

Se si verifica il primo caso, si mostri secondo quale ordine i processi possano essere terminati. Cosa accadrebbe se il vettore di richieste relativo al secondo processo fosse  $[2, 1, 0, 1]$ ?

*Soluzione*

Il vettore *Available* (che può essere calcolato come differenza fra il vettore totale delle risorse,  $[4,2,3,1]$ , ed il vettore delle risorse attualmente allocate,  $[2,1,3,1]$ ) risulta:

$$Available = \begin{bmatrix} 2 & 1 & 0 & 0 \end{bmatrix}.$$

Conseguentemente la successione di processi  $\langle P_3, P_2, P_1 \rangle$  (che produce i vettori *Available*  $[2,2,2,0]$ ,  $[4,2,2,1]$  e  $[4,2,3,1]$ ) è una successione sicura  $\Rightarrow$  il sistema è in stato sicuro.

Viceversa, se  $Need_2 = [2, 1, 0, 1]$  lo stato del sistema diventa non sicuro, dato che l'unico processo le cui richieste possono essere assecondate è il processo  $P_3$ , mentre  $P_1$  e  $P_2$  potrebbero venir coinvolti in un deadlock.

- **Esercizio 5**

Quattro processi,  $P_1, P_2, P_3, P_4$ , competono per l'uso di un certo numero di risorse dei tipi A, B, C. La disponibilità di risorse nel sistema è la seguente: 16 istanze di A, 8 di B e 10 di C. Supponendo che le quantità massime di risorse richieste da ogni processo siano:

$$Max = \begin{bmatrix} 8 & 5 & 6 \\ 4 & 2 & 5 \\ 6 & 6 & 4 \\ 5 & 4 & 7 \end{bmatrix}$$

e le risorse allocate ad un certo istante siano:

$$Allocation = \begin{bmatrix} 5 & 0 & 3 \\ 2 & 1 & 2 \\ 4 & 3 & 2 \\ 3 & 0 & 0 \end{bmatrix},$$

controllare se lo stato è *sicuro* e, in tal caso, se la richiesta  $[0, 1, 1]$  di  $P_2$  può essere soddisfatta.

*Soluzione*

Il vettore *Available* e la matrice *Need* valgono rispettivamente

$$Available = \begin{bmatrix} 2 & 4 & 3 \end{bmatrix}, \quad Need = \begin{bmatrix} 3 & 5 & 3 \\ 2 & 1 & 3 \\ 2 & 3 & 2 \\ 2 & 4 & 7 \end{bmatrix}.$$

Di conseguenza, può essere servito subito il processo  $P_2$  che, in seguito, rilascia le risorse precedentemente possedute, producendo  $Available = [4, 5, 5]$ ; ora può essere servito  $P_1$  che, al termine della sua esecuzione, fornisce  $Available = [9, 5, 8]$ . A questo punto si può procedere in sequenza con  $P_3$  e  $P_4$ . Quindi lo stato è sicuro.

Se viene accordata la nuova richiesta del processo  $P_2$ , *Available* e *Need* divengono rispettivamente,

$$Available = \begin{bmatrix} 2 & 3 & 2 \end{bmatrix}, \quad Need = \begin{bmatrix} 3 & 5 & 3 \\ 2 & 0 & 2 \\ 2 & 3 & 2 \\ 2 & 4 & 7 \end{bmatrix}.$$

In questo caso, può ancora essere servito  $P_2$  e, in seguito, gli altri processi, per esempio secondo la sequenza descritta precedentemente.

• **Esercizio 6**

Cinque processi,  $P_0, P_1, P_2, P_3, P_4$ , condividono un insieme di risorse di quattro tipi diversi, A, B, C, D. Si supponga di trovarsi nella configurazione seguente:

$$Allocation = \begin{bmatrix} 3 & 1 & 4 & 5 \\ 6 & 0 & 2 & 3 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 4 & 1 \end{bmatrix}, \quad Max = \begin{bmatrix} 3 & 2 & 7 & 5 \\ 6 & 0 & 2 & 6 \\ 7 & 1 & 0 & 5 \\ 0 & 0 & 2 & 1 \\ 6 & 0 & 6 & 5 \end{bmatrix}, \quad Available = \begin{bmatrix} 5 & 1 & 0 & 2 \end{bmatrix}.$$

Usando l'algoritmo del Banchiere, rispondere alle seguenti domande:

- Lo stato è *sicuro*?
- Se è sicuro, enumerare tutte e sole le sequenze sicure.
- Se in questa configurazione il processo  $P_2$  avanza una richiesta per  $[4, 0, 0, 2]$ , tale richiesta può essere soddisfatta immediatamente?

*Soluzione*

- Lo stato è sicuro; infatti è possibile trovare una sequenza di processi sicura, cioè tale che per ogni processo le richieste non ancora soddisfatte possono essere soddisfatte dalle risorse disponibili più le risorse attualmente in uso ai processi che lo precedono nella sequenza. In questo caso, la matrice *Need* è:

$$Need = \begin{bmatrix} 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 3 \\ 6 & 1 & 0 & 5 \\ 0 & 0 & 0 & 0 \\ 6 & 0 & 2 & 4 \end{bmatrix}.$$

Pertanto,  $P_3$  ha attualmente allocate tutte le risorse necessarie; quando  $P_3$  termina il vettore *Available* diviene  $[5, 1, 2, 3]$ . A questo punto può essere servito il processo  $P_1$  che, una volta eseguito, rilascia  $Available = [11, 1, 4, 6]$ . In seguito, i rimanenti processi  $P_0, P_2$  e  $P_4$  possono essere eseguiti in un ordine qualsiasi.

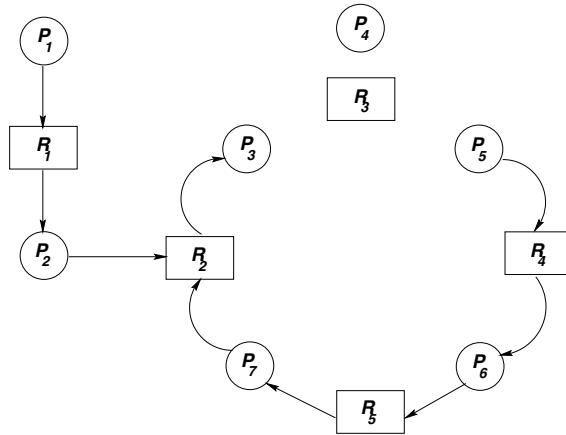
- Per quanto detto al punto precedente,  $\langle P_3, P_1, P_0, P_4, P_2 \rangle, \langle P_3, P_1, P_0, P_2, P_4 \rangle, \langle P_3, P_1, P_2, P_0, P_4 \rangle, \langle P_3, P_1, P_4, P_0, P_2 \rangle, \langle P_3, P_1, P_4, P_2, P_0 \rangle$  sono tutte e sole le sequenze sicure.
- La richiesta non può essere immediatamente soddisfatta in quanto la configurazione in cui il sistema verrebbe a trovarsi se fosse soddisfatta,

$$Allocation = \begin{bmatrix} 3 & 1 & 4 & 5 \\ 6 & 0 & 2 & 3 \\ 5 & 0 & 0 & 2 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 4 & 1 \end{bmatrix}, \quad Available = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix},$$

non rappresenta uno stato sicuro. Infatti dopo che è stato eseguito  $P_3$ , le risorse disponibili, rappresentate dal vettore *Available* =  $[1, 1, 2, 1]$ , non sono sufficienti a soddisfare le richieste massime relative a nessun processo. (Si noti che la matrice *Need* è identica a quella precedentemente descritta, eccetto che per la riga relativa al processo  $P_2$  che adesso è  $[2, 1, 0, 3]$ ).

- **Esercizio 7**

Si consideri il seguente grafo di allocazione di risorse:



e sia  $Max$  tale che

$$Max = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Se la tecnica prescelta per gestire i deadlock consiste nell'*evitarli*, una richiesta della risorsa  $R_3$  da parte di  $P_5$  può essere soddisfatta? Motivare la risposta.

*Soluzione*

La richiesta di  $P_5$  non può essere soddisfatta perché altrimenti lo stato del sistema diventa non sicuro. Infatti, la matrice  $Max$  suggerisce che anche il processo  $P_3$  potrebbe richiedere  $R_3$  e, in questo caso, si verificherebbe una attesa circolare (corrispondente ad un ciclo nel grafo di allocazione delle risorse). I processi coinvolti nel deadlock sarebbero  $P_3$ ,  $P_5$ ,  $P_6$  e  $P_7$  e le risorse  $R_2$ ,  $R_3$ ,  $R_4$  ed  $R_5$ .

- **Esercizio 8**

Si consideri un sistema costituito da 5 processi,  $P_0, P_1, P_2, P_3, P_4$ , e dalle 4 classi di risorse di tipo A, B, C, D, descritto dalle seguenti matrici

$$Allocation = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 1 & 0 & 0 & 0 \\ 1 & 3 & 5 & 4 \\ 0 & 6 & 3 & 2 \\ 0 & 0 & 1 & 4 \end{bmatrix}, \quad Max = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 1 & 7 & 5 & 0 \\ 2 & 3 & 5 & 6 \\ 0 & 6 & 5 & 2 \\ 0 & 6 & 5 & 6 \end{bmatrix}, \quad Available = \begin{bmatrix} 1 & 5 & 2 & 0 \end{bmatrix}.$$

Utilizzando l'*algoritmo del Banchiere* si stabilisca se il sistema è in stato sicuro e, in caso affermativo, se lo stato si mantiene sicuro a fronte di una ulteriore richiesta  $[0, 4, 2, 0]$ , da parte del processo  $P_1$ .

*Soluzione*

In questo caso, la matrice  $Need = Max - Allocation$  risulta:

$$Need = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 7 & 5 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 2 & 0 \\ 0 & 6 & 4 & 2 \end{bmatrix}.$$

Pertanto la successione  $< P_0, P_2, P_1, P_3, P_4 >$  ( $Available = [1, 5, 3, 2] \rightarrow [2, 8, 8, 6] \rightarrow [3, 8, 8, 6] \rightarrow [3, 14, 11, 8] \rightarrow [3, 14, 12, 12]$ ) è una successione sicura e tale è anche lo stato descritto dalle matrici  $Available$ ,  $Max$  e dal vettore  $Available$ .

Nel caso in cui venga accordata al processo  $P_1$  l'ulteriore richiesta di risorse  $[0, 4, 2, 0]$  le matrici  $Allocation$  e  $Need$  divengono:

$$Allocation = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 1 & 4 & 2 & 0 \\ 1 & 3 & 5 & 4 \\ 0 & 6 & 3 & 2 \\ 0 & 0 & 1 & 4 \end{bmatrix}, \quad Need = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 3 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 2 & 0 \\ 0 & 6 & 4 & 2 \end{bmatrix},$$

mentre il vettore delle risorse disponibili risulta  $Available = [1, 1, 0, 0]$ . Può quindi essere ancora una volta servito il processo  $P_0$  che restituisce le proprie risorse, fornendo  $Available = [1, 1, 1, 2]$ . Di nuovo, può essere servito il processo  $P_2$ , che produce  $Available = [2, 4, 6, 6]$ . In seguito, vengono serviti i processi  $P_1$  ( $Available = [3, 8, 8, 6]$ ),  $P_3$  ( $Available = [3, 14, 11, 8]$ ) e  $P_4$  ( $Available = [3, 14, 12, 12]$ ). Pertanto, la stessa sequenza  $< P_0, P_2, P_1, P_3, P_4 >$  è sicura e tale è anche lo stato del sistema.

## Esercizi da svolgere

1. Un sistema è dotato delle seguenti 4 risorse: 4 unità a nastro, 2 plotter, 3 stampanti e una unità CD-ROM. Tre processi,  $P_1$ ,  $P_2$  e  $P_3$  sono in esecuzione. La matrice  $Max$  rappresenta le loro richieste massime, e la matrice  $Allocation$  l'allocazione attuale (le risorse sono considerate nell'ordine sopra detto).

$$Allocation = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \end{bmatrix} \quad Max = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 3 & 0 & 1 & 1 \\ 2 & 2 & 2 & 0 \end{bmatrix}$$

Si stabilisca se lo stato è sicuro o non sicuro. Se lo stato è sicuro, si mostri secondo quale ordine i processi possano essere terminati.

Si ripeta l'analisi nel caso che il processo  $P_2$  abbia un vettore di richiesta massima  $Max_2 = [4 \ 1 \ 0 \ 2]$ .

2. Considerando l'algoritmo del Banchiere di Dijkstra si stabilisca quando gli stati seguenti sono sicuri o non sicuri. Se uno stato è sicuro, si mostri secondo quale ordine i processi possano essere terminati. Se uno stato è non sicuro, si mostri in che modo sia possibile evolvere verso una situazione di deadlock.

Processo no.	Risorse allocate	Risorse max	Risorse disponibili
$P_0$	2	6	1
$P_1$	4	7	
$P_2$	5	6	
$P_3$	0	2	

Processo no.	Risorse allocate	Risorse max	Risorse disponibili
$P_0$	4	8	2
$P_1$	3	9	
$P_2$	5	8	