



# Sistemi Operativi

## Esercizi Sincronizzazione

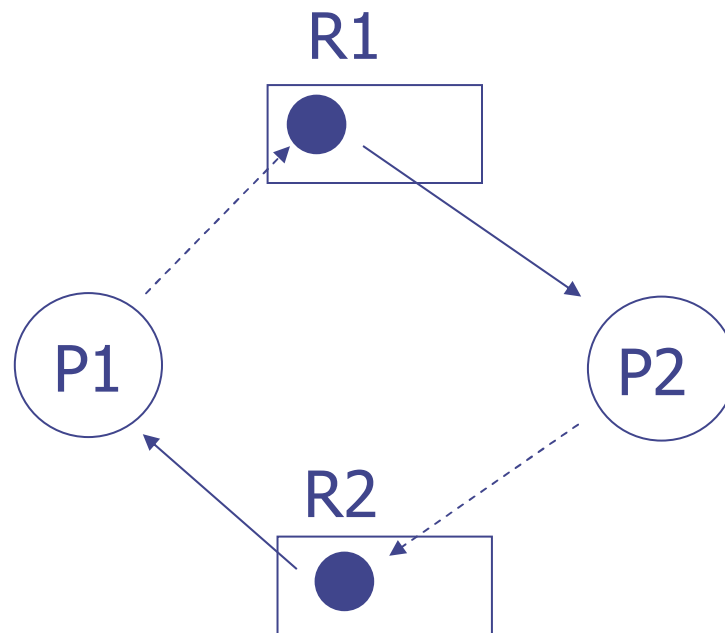
Docente: Claudio E. Palazzi  
[cpalazzi@math.unipd.it](mailto:cpalazzi@math.unipd.it)

# Grafo di Assegnazione delle Risorse

- ◆ Si consideri un sistema con 2 processi (P1, P2), e 2 tipologie di risorse (R1, R2) con disponibilità: 1 risorsa di tipo R1 e 1 risorsa di tipo R2.
- ◆ Si consideri la seguente cronologia di richieste:
  - 1) P2 richiede R1
  - 2) P1 richiede R2
  - 3) P2 richiede R2
  - 4) P1 richiede R1
- ◆ Verificare se alla fine il sistema si trovi in condizioni di stallo

# Grafo di Assegnazione delle Risorse

◆ Il grafo di allocazione delle risorse sarà il seguente:



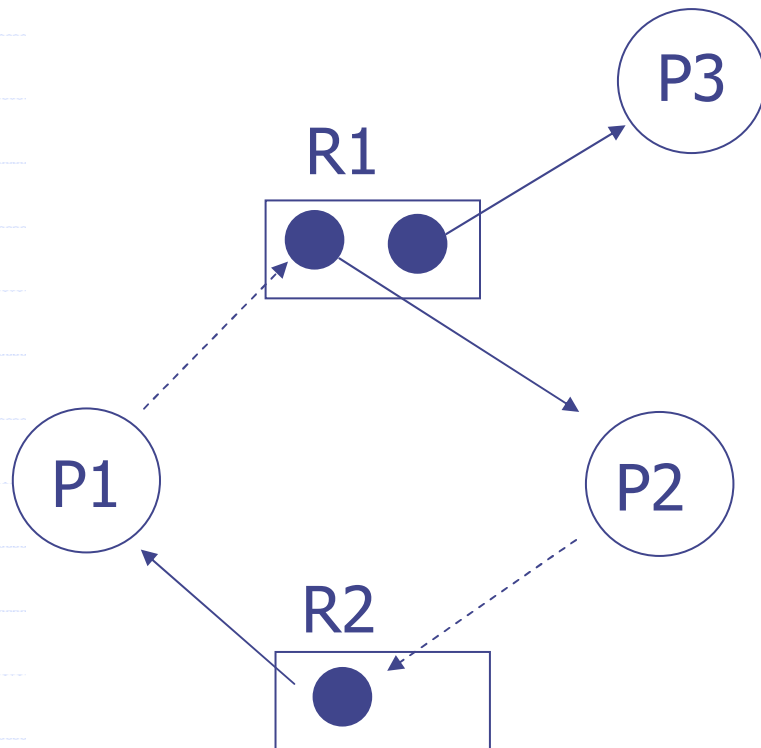
Si forma un ciclo:  
dunque il sistema è  
in **stallo**

# Grafo di Assegnazione delle Risorse

- ◆ Si consideri un sistema con 3 processi (P1, P2, P3), e 2 tipologie di risorse (R1, R2) con disponibilità: 2 risorse di tipo R1 e 1 risorsa di tipo R2.
- ◆ Si consideri la seguente cronologia di richieste:
  - 1) P2 richiede R1
  - 2) P1 richiede R2
  - 3) P2 richiede R2
  - 4) P3 richiede R1
  - 5) P1 richiede R1
- ◆ Verificare se alla fine il sistema si trovi in condizioni di stallo

# Grafo di Assegnazione delle Risorse

◆ Il grafo di allocazione delle risorse sarà il seguente:



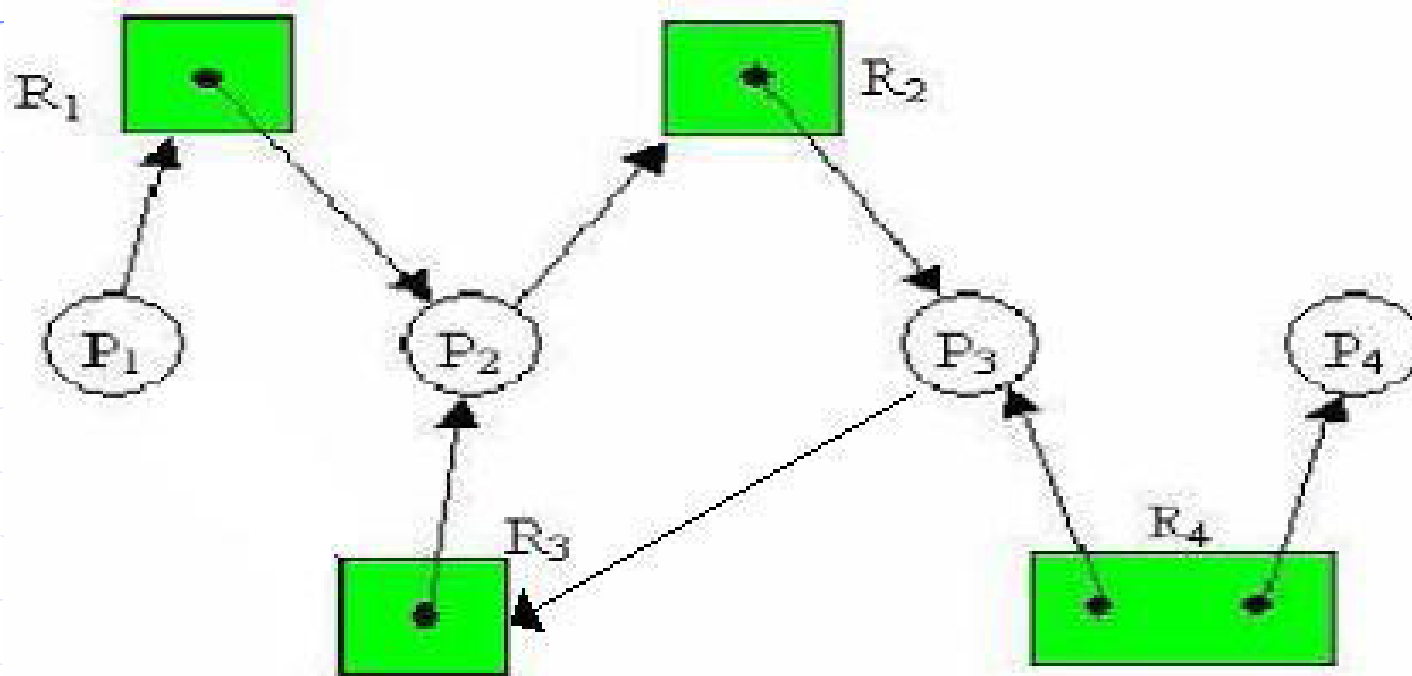
Si forma un ciclo ma il sistema NON è in **stallo**

Infatti un elemento del ciclo (R1) ha molteplicità 2 e una delle due istanze è fuori da cicli: P3 può terminare e rilasciare una istanza di R1 che viene così assegnata a P1

# Verifica Deadlock

- ◆ Si consideri un sistema con 4 processi (P1, P2, P3, P4), e 4 tipologie di risorse (R1, R2, R3, R4) con disponibilità: 1 risorsa di tipo R1, 1 risorsa di tipo R2, 1 risorsa di tipo R3, 2 risorse di tipo R4.
- ◆ Si assuma che:
  - ogni volta che un processo richieda una risorsa libera, questa venga assegnata al processo richiedente;
  - ogni volta che un processo richieda una risorsa già occupata, il richiedente deve attendere che la risorsa si liberi prima di impossessarsene (coda FIFO di attesa)
- ◆ Si consideri la seguente successione cronologica di richieste e rilasci di risorse:
  - 1) P2 richiede R1, R2, R3
  - 2) P3 richiede R2, R4
  - 3) P2 rilascia R2
  - 4) P4 richiede R4
  - 5) P1 richiede R1
  - 6) P2 richiede R2
  - 7) P3 richiede R3
- ◆ Verificare se alla fine il sistema si trovi in condizioni di stallo

# Sol – Grafo Allocazione Risorse



- ◆ Alla fine delle operazioni descritte, il grafo di allocazione delle risorse appare come in figura. In tale grafo esiste un ciclo di richieste/assegnazioni che coinvolge P<sub>2</sub>, R<sub>2</sub>, P<sub>3</sub>, R<sub>3</sub>: pertanto, il sistema **è in stallo**.

# Verifica Deadlock

- ◆ Dato il sistema descritto dalla seguente rappresentazione insiemistica di assegnazione delle risorse:

$P = P_1, P_2, P_3, P_4$

$R = R_1, R_2, R_2, R_3$

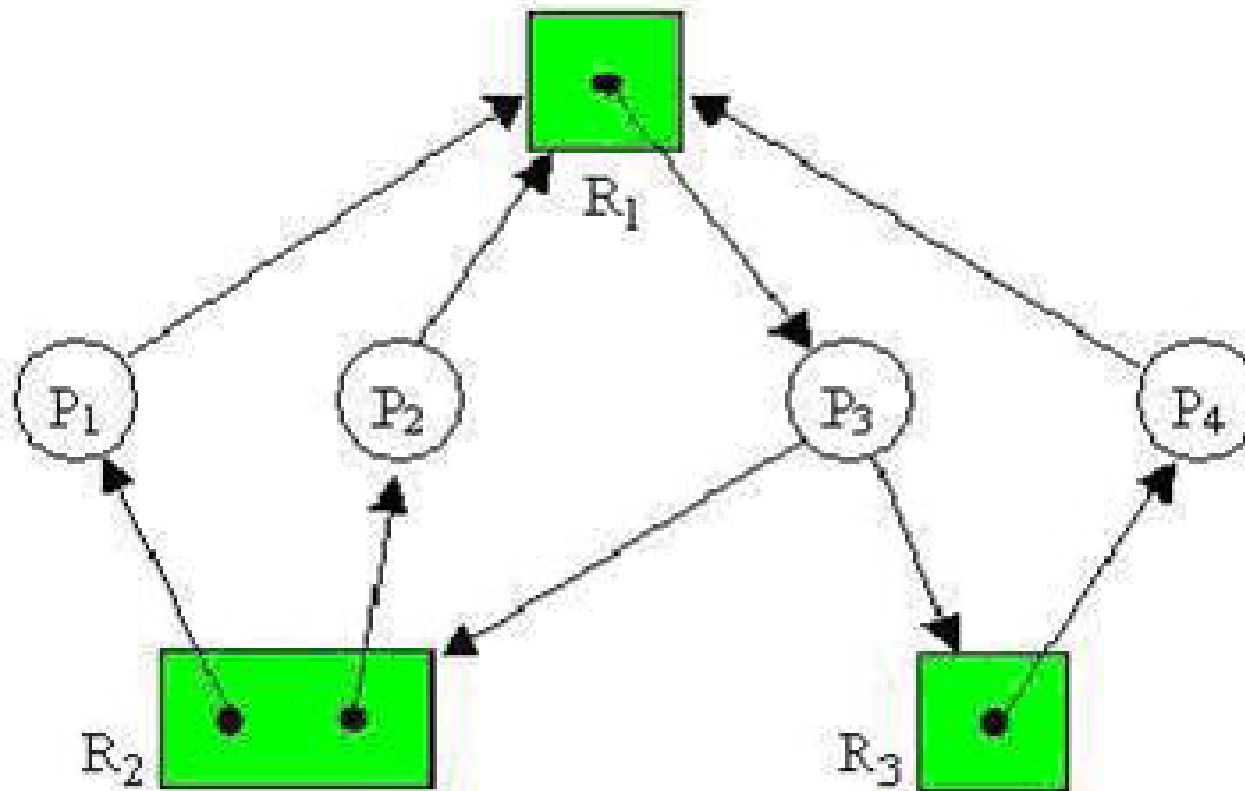
$E = P_1 \rightarrow R_1, P_2 \rightarrow R_1, P_3 \rightarrow R_2, P_3 \rightarrow R_3, P_4 \rightarrow R_1$

$R_1 \rightarrow P_3, R_2 \rightarrow P_1, R_2 \rightarrow P_2, R_3 \rightarrow P_4$

- ◆ si verifichi, anche tramite analisi del relativo grafo di allocazione delle risorse, se il sistema si trovi in condizione di stallo.
- ◆ Successivamente, si discuta per ogni processo se la sua rimozione forzata (con conseguente liberazione delle risorse eventualmente in suo possesso ed annullamento delle sue richieste di risorse) porti ad un cambiamento della situazione precedentemente rilevata.



# Sol – Grafo Allocazione Risorse



# Sol – Grafo Allocazione Risorse

- ◆ La figura riporta la versione grafica della rappresentazione insiemistica data, al cui interno si distinguono 3 percorsi chiusi:

percorso 1:  $R_2 \rightarrow P_1 \rightarrow R_1 \rightarrow P_3 \rightarrow R_2$

percorso 2:  $R_2 \rightarrow P_2 \rightarrow R_1 \rightarrow P_3 \rightarrow R_2$

percorso 3:  $R_3 \rightarrow P_4 \rightarrow R_1 \rightarrow P_3 \rightarrow R_3$

- ◆ I percorsi 1 e 2 contengono risorse a molteplicità  $> 1$ , e quindi per ciascuno di essi occorre una specifica analisi di dettaglio. Il percorso 3 invece contiene solo risorse unarie, per cui possiamo affermare che i processi P3 e P4 sono sicuramente in stallo. Essendo il processo P3, che è in stallo, presente anche nei percorsi 1 e 2, possiamo concludere che anche i processi P1 e P2 si trovano in situazione di stallo. Pertanto, in definitiva, l'intero sistema è in **stato di stallo**.

# Sol – Grafo Allocazione Risorse

◆ Analizziamo ora la situazione che seguirebbe all'eliminazione di singoli processi dal sistema:

- eliminiamo P1: la situazione non cambierebbe in quanto l'anello di sicuro stallo (percorso 3) non verrebbe intaccato; inoltre anche il processo P2 rimarrebbe bloccato dato che la risorsa R1 da esso richiesta è impegnata nel percorso 3.
- eliminiamo P2: la situazione è analoga alla precedente: gli altri processi rimarrebbero bloccati.
- eliminiamo P3: la situazione cambierebbe, dato che R1, posseduta dal processo P3 eliminato, diverrebbe libera; i processi P1, P2 e P4 hanno tutti una richiesta pendente per tale risorsa, ed indipendentemente dall'ordine con il quale possano venir soddisfatte tali richieste, i processi riprenderanno sequenzialmente ad avanzare, conseguentemente liberando il sistema dal precedente stato di stallo.
- eliminiamo P4: la situazione non cambierebbe in quanto la risorsa R3, rilasciata dal processo eliminato, verrebbe assegnata al processo P3; che però resterebbe ancora bloccato stante la perdurante indisponibilità della risorsa R2, con il conseguente blocco dei percorsi 1 e 2.

# Verifica Stato Sicuro

- ◆ Un sistema si compone di 4 processi e 5 risorse condivise a diversa molteplicità.
- ◆ All'istante considerato, lo stato di allocazione delle risorse a processi e i loro bisogni massimi previsti sono riportati in Tabella.

Processo	Risorse ( $R_1, R_2, R_3, R_4, R_5$ )				
	Allocazione attuale		Richiesta massima		
A	1	0	2	1	1
B	2	0	1	1	1
C	1	1	0	1	0
D	1	1	1	1	0

- ◆ Assumendo che il vettore di disponibilità delle risorse allo stato corrente sia uguale a  $[0 \ 0 \ 2 \ 1 \ 2]$ , si discuta se il sistema sia in *deadlock*.

# Verifica Deadlock – Sol

- ◆ Vettore disponibilità [0 0 2 1 2]

Processo	Risorse ( $R_1, R_2, R_3, R_4, R_5$ )				
	Allocazione attuale		Richiesta massima		
A	1	0	2	1	1
B	2	0	1	1	1
C	1	1	0	1	0
D	1	1	1	1	0

- ◆ il processo D può essere eseguito fino alla fine
- ◆ Quando ha finito, il vettore delle risorse disponibili è [1 1 3 2 2]
- ◆ Sfortunatamente però ora il sistema si troverebbe in **deadlock**
- ◆ Dunque la situazione iniziale non rappresentava uno **stato sicuro**