Corso di Programmazione Concorrente

Stallo

Valter Crescenzi
crescenz@dia.uniroma3.it
http://www.dia.uniroma3.it/~crescenz

Assunzione di Progresso Finito

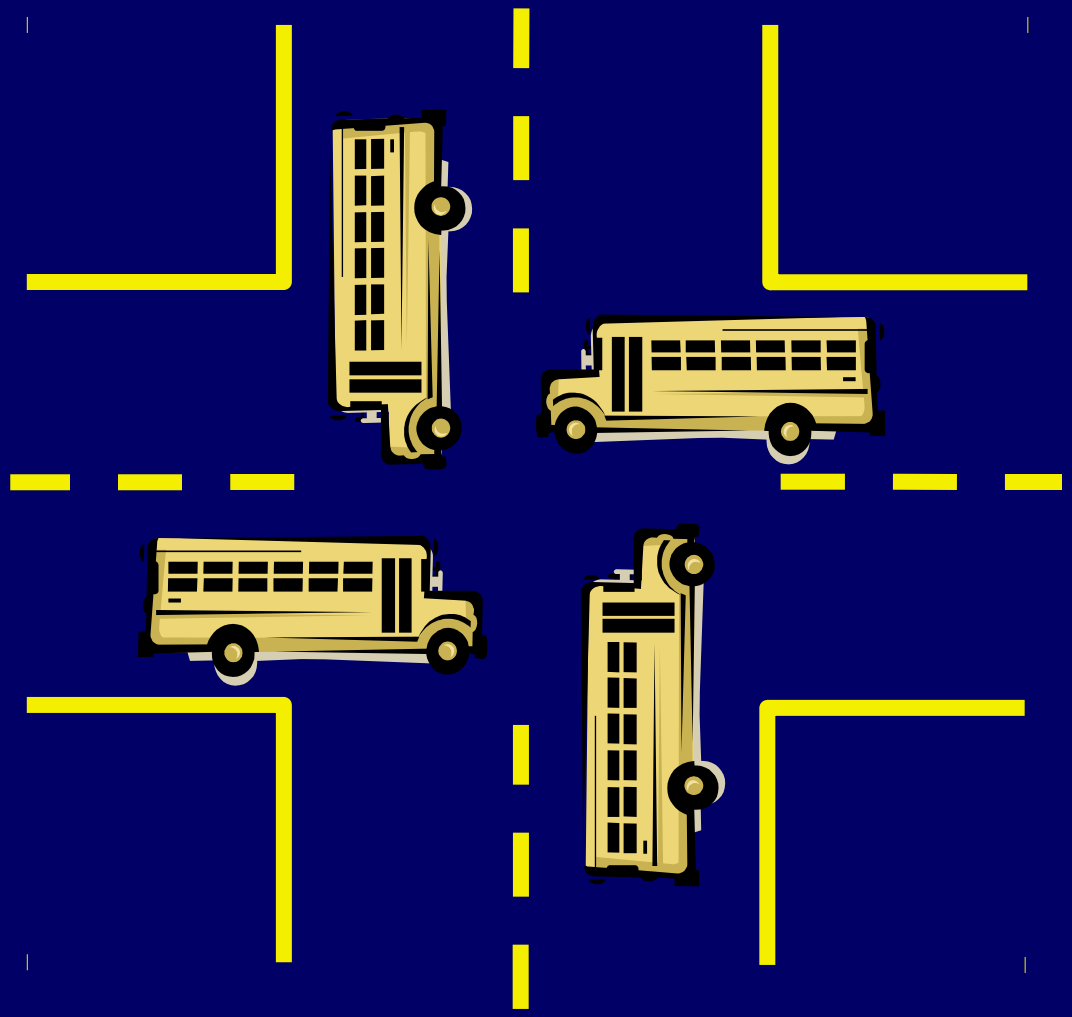
Tutti i processori virtuali hanno una velocità finita non nulla

- Questa assunzione è l'unica che si può fare sui processori virtuali e sulle loro velocità relative

Starvation & Deadlock

- Esistono due diverse situazioni che possono invalidare l'assunzione di progresso finito
  - *Starvation* (o *live-lock*): quando un f.d.e. rimane in attesa di un evento che pure si verifica infinite volte
    - un sistema di f.d.e. che garantisce contro questa evenienza si dice che gode della proprietà di *fairness*
  - *Deadlock* (o *stallo*): quando due o più f.d.e. rimangono in attesa di eventi che non potranno mai verificarsi a causa di condizioni cicliche nel possesso e nella richiesta di risorse
    - esempio classico: un f.d.e. P_1 possiede una risorsa R_a e richiede una risorsa R_b già posseduta da un altro f.d.e. P_2; quest'ultimo a sua volta richiede l'uso di R_a

Esempio di Stallo



- Nessun pulmino può completare l’attraversamento
- Ciascuno pulmino attende che il pulmino che gli blocca la strada completi l’attraversamento e liberi l’incrocio
- Sono tutti in attesa di un evento che non può verificarsi

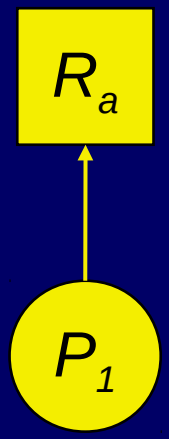
Condizioni per lo Stallo (di Coffman)

- Affinché si manifesti uno stallo, devono verificarsi *tutte* le seguenti condizioni
  - *Condizione di* mutua-esclusione
    - le risorse condivise sono seriali
  - *Condizione di* incrementalità delle richieste
    - i processi richiedono le risorse una alla volta
  - *Condizione di* non-prerilasciabilità
    - le risorse non sono prerilasciabili
  - *Condizione di* attesa circolare
    - >>

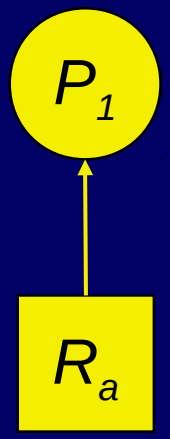
Grafo di Allocazione delle Risorse (Holt)

- Un formalismo molto semplice per rappresentare la situazione di più f.d.e. e delle risorse che richiedono
- N.B. Utile in caso di risorse seriali

- *P_1 richiede R_a*

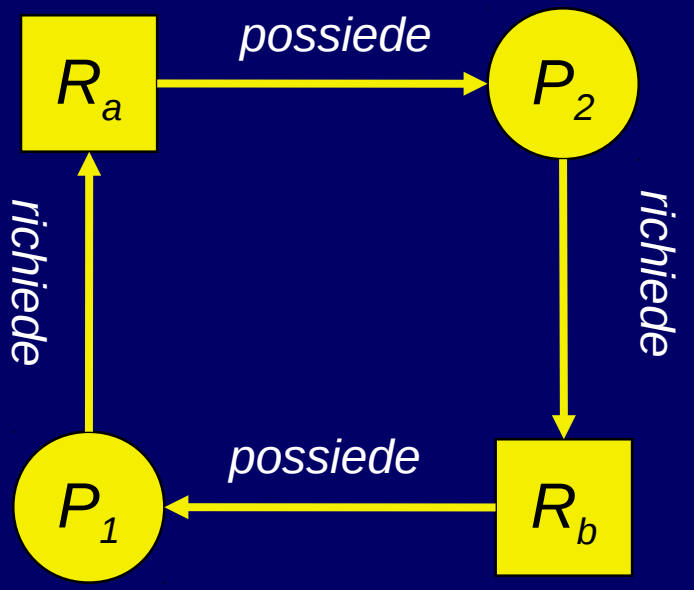


- *P_1 possiede R_a*

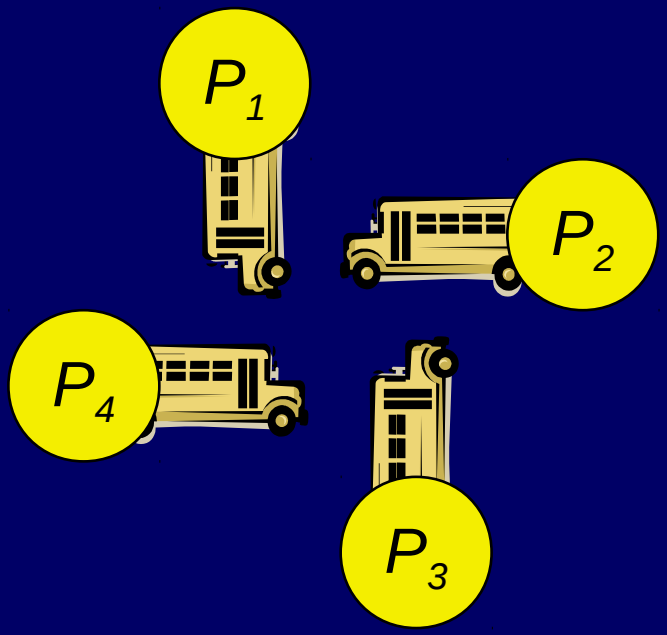


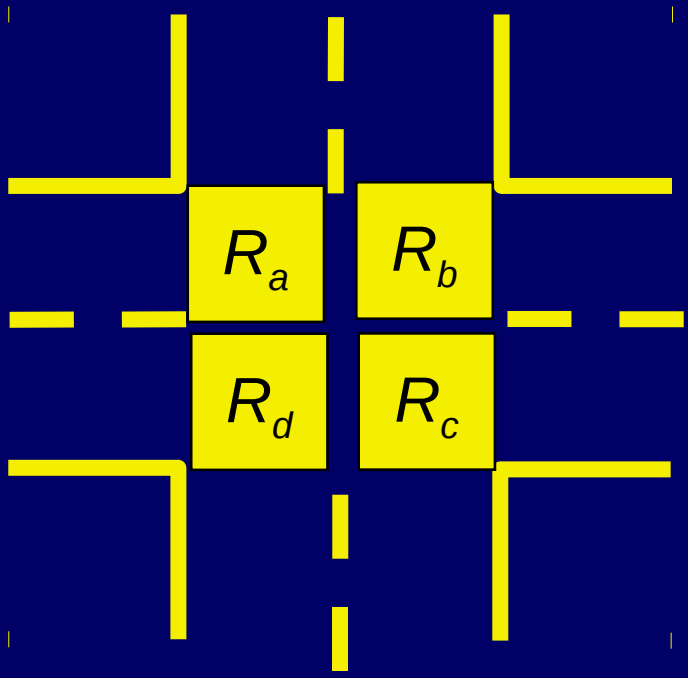
Identificazione dello Stallo

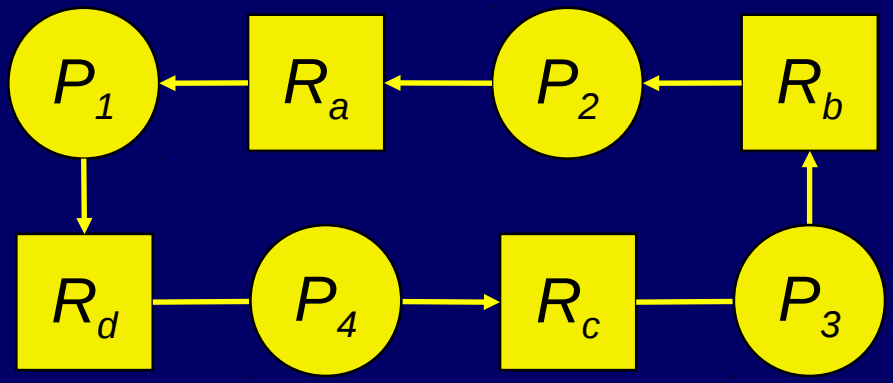
- Dato un grafo di allocazione delle risorse, lo stallo è in atto quando il grafo contiene dei cicli
- Attesa circolare tra due f.d.e. e due risorse



Esempio di Attesa Circolare







Tecniche di Gestione dello Stallo

- Fuga dallo stallo
  - si rifiutano le richieste di risorse che causano o potrebbero causare lo stallo
  - >> Algoritmo del Banchiere
- Riconoscimento e risoluzione dello stallo
  - si analizza il grafo di allocazione delle risorse per rilevare eventuali stalli in atto
  - quindi tecniche di risoluzione
- Prevenzione dello stallo
  - si cerca di prevenire lo stallo invalidando una delle condizioni di Coffman

Fuga dallo Stallo

- Strategia conservativa: si ammette la possibilità di rifiutare (posticipare) alcune richieste
- Consiste nel mantenersi uno stato *sicuro* nell'allocazione delle risorse
- Lo stato è *sicuro* se
  - non si è in stallo
  - si è in grado di soddisfare in un tempo finito *tutte* le richieste non ancora evase
- Dijkstra propose un algoritmo di fuga dello stallo noto come Algoritmo del Banchiere

Algoritmo del Banchiere

- Facciamo riferimento al caso in cui si gestisce una sola tipologia di risorse indivisibili (pagine di memoria, stampanti, euro)
- generalizzabile al caso di più tipologie di risorse

- Un banchiere (gestore della risorsa) concede prestiti (risorse) ai propri clienti (f.d.e.)

Algoritmo del Banchiere (Singola Risorsa)

- Regole osservate dal Banchiere
  - ogni cliente dichiara in anticipo il proprio fido
  - la cassa iniziale della banca è capace di coprire il massimo fido
  - i clienti si impegnano a restituire il prestito in un tempo finito, ma possono aver bisogno di raggiungere il tetto del fido prima di cominciare a farlo
  - il banchiere può temporaneamente rifiutare un prestito se ritiene che in seguito alla concessione potrebbe non riuscire a garantire in un tempo finito il fido di tutti i clienti

Esempio: Algoritmo del Banchiere (1)

	Prestito attuale	Fido	Potenziale Richiesta
Cliente A	8	10	2
Cliente B	4	8	4
Cliente C	3	6	3

Cassa = 2

- Il banchiere è in uno stato sicuro perché può soddisfare le potenziali richieste di A
- Quando questo termina sarà possibile (con i 10 rientrati dal suo fido) soddisfare gli altri

Esempio: Algoritmo del Banchiere (2)

	Prestito attuale	Fido	Potenziale Richiesta
Cliente A	8	10	2
Cliente B	4	8	4
Cliente C	3	6	3

Cassa = 2

- Se A chiede uno od anche due, il banchiere concorda il prestito. Infatti…

Esempio: Algoritmo del Banchiere (3)

	Prestito attuale	Fido	Potenziale Richiesta
Cliente A	10	10	0
Cliente B	4	8	4
Cliente C	3	6	3

Cassa = 0

- Da questo momento A, in un tempo finito, può solo restituire il prestito

Esempio: Algoritmo del Banchiere (4)

	Prestito attuale	Fido	Potenziale Richiesta
Cliente A	8	10	2
Cliente B	4	8	4
Cliente C	3	6	3

Cassa = 2

- Se C chiede uno, il banchiere rifiuta il prestito. Altrimenti…

Esempio: Algoritmo del Banchiere (5)

	Prestito attuale	Fido	Potenziale Richiesta
Cliente A	8	10	2
Cliente B	4	8	4
Cliente C	4	6	2

Cassa = 1

- Stato non sicuro: non può più soddisfare l'eventuale richiesta di alcun cliente
- Pericolo di stallo: tutti possono cominciare a chiedere ma nessuno restituisce

Esempio: Algoritmo del Banchiere (6)

	Prestito attuale	Fido	Potenziale Richiesta
Cliente A	1	2	1
Cliente B	4	8	4
Cliente C	3	6	3

Cassa = 1

- Altro stato non sicuro: anche se A può concludere il prestito con l'unità in cassa, il suo fido non basta a coprire le potenziale richieste degli altri clienti

Problemi della Tecnica Basata sulla Fuga dallo Stallo

- Il problema principale è che si tratta di un approccio *cooperativo* e *conservativo*:
  - *cooperativo*: i processi devono specificare in anticipo il numero massimo di unità di risorse che intendono consumare
  - *conservativo*: pur di non correre il pericolo di stallo, vengono rifiutate richieste che non necessariamente conducono allo stallo; alcune risorse potrebbero risultare sotto-utilizzate

Esercizio: implementare l'algoritmo del banchiere.

Riconoscimento e Risoluzione dello stallo

- Strategia ottimistica: non si prendono precauzioni contro lo stallo
- Si rilevano le situazioni di stallo che possono verificarsi
  - ad esempio cercando cicli nel grafo di allocazione delle risorse
- Quindi, risoluzione dello stallo
  - eliminando uno dei f.d.e. coinvolti e causandone il rilascio forzato delle risorse
  - tecniche di rollback: alcuni esecutori possono supportare il disfacimento delle operazioni di un f.d.e.

Prevenzione dello Stallo

- Invalidiamo una delle condizioni di Coffman alla volta
  - *Condizione di* mutua-esclusione
    - le risorse condivise sono seriali
  - *Condizione di* incrementalità delle richieste
    - i f.d.e. richiedono le risorse incrementalmente
  - *Condizione di* non-prerilasciabilità
    - le risorse non sono prerilasciabili
  - *Condizione di* attesa circolare
- Per la prima c'è poco da fare: la serialità è una caratteristica intrinseca della risorsa

Invalidare l’Incrementalità delle Richieste

- Le risorse utilizzate vengono richieste atomicamente: o si ottengono tutte o nessuna
- Semplice, ma almeno due grandi svantaggi:
  - è necessario la cooperazione dei f.d.e. che devono richiedere tutte le risorse in anticipo
  - si rischia che una risorsa risulta utilizzata, complessivamente, solo per una piccola frazione del tempo in cui rimane allocata al f.d.e.

Invalidare la Non-Prerilasciabilità

- Tutte le volte che un f.d.e. si vede rifiutare una richiesta di risorsa, deve rilasciare tutte quelle già ottenute e ricominciare
- Svantaggi:
  - costi intrinseci del prerilascio forzato
  - degrado di rendimento complessivo
  - non applicabile a risorse non prerilasciabili

Invalidare l'Attesa Circolare

- Si adotta uno schema di allocazione delle risorse che impedisce la creazione di cicli nel grafo di allocazione
- Allocazione gerarchica delle risorse:
  - si impone un ordinamento totale delle risorse ad es.: $R_a<R_b<R_c<R_d<R_e<\ldots$
  - i f.d.e. devono richiedere le risorse rispettandone l'ordinamento gerarchico
- Lo svantaggio principale è la possibile sottoutilizzazione delle risorse

Un pò di Sano Realismo

- Come nel caso più generico della gestione dell'interferenza, anche per la gestione dello stallo la strategia più conveniente dipende ampiamente dal contesto di riferimento
  - scenario più comune
    - gli stalli sono rari
    - i costi delle tecniche di fuga, rilevamento, e prevenzione dello stallo sono ritenuti inaccettabili
    - strategia che va per la maggiore: *non fare nulla*
  - scenari più particolari
    - lo stallo ha conseguenze inaccettabili
    - l'esecutore supporta il disfacimento delle operazioni
      - risultano applicabili anche strategie più sofisticate