SISTEMI OPERATIVI

02.b

La concorrenza e le risorse

Concorrenza: risorse

- Le Risorse
- Evoluzione di processi
  - diagrammi di stato notevoli
- Stallo o Blocco Critico
  - individuazione dello stallo
  - prevenzione dello stallo
  - algoritmo del banchiere

DEI UNIV PD © 2005

Le Risorse

- Come per il processo, anche la risorsa è una astrazione.

- **Risorsa** = qualunque entità necessaria ad un processo per sviluppare la propria attività.

Indisponibilità di una risorsa → Blocco dell'evoluzione del processo che la richiede

- Ad esempio perché in dotazione ad un altro processo
  Sottrazione forzata di una risorsa = preemption.

Le Risorse [2]

- Il concetto di **risorsa** può essere applicato anche al mondo reale

Mondo reale → **Sistemi di Elaborazione**

- Auto per andare a lavorare
- Dentifricio
- Strumenti e attrezzi da lavoro
- :

- Stampante
- Dischi magnetici
- Dati prodotti da altri
- :

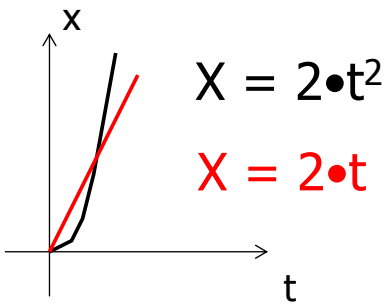
Attributi

EQUIVALENTI → NON EQUIVALENTI
CONSUMABILI → NON CONSUMABILI
CONDIVISIBILI → MUTUAMENTE ESCLUSIVE

Proprietà delle risorse

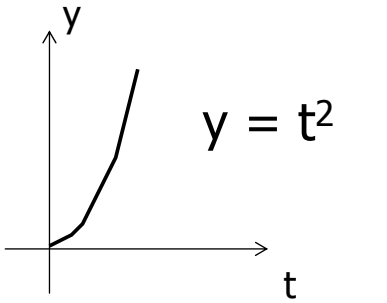
- Consumabili — Distingue tra risorse che possono o no essere riutilizzate. Es. un buffer condiviso fra produttore e consumatore è riutilizzabile; un dato estratto dal consumatore non lo è.
- Condivisibili — Possono essere usate da più di un processo contemporaneamente. Es. il disco magnetico
- Non condivisibili — Occorre attendere che un processo ne abbia terminato l'utilizzo prima che un altro processo possa ottenerne il possesso.
- Sottraibili — Possibilità per le risorse di essere sottratte ai processi che ne stanno usufruendo. Es. il processore è sottraibile (se parto dal presupposto che posso salvare il contesto...).

Rappresentazione di un moto

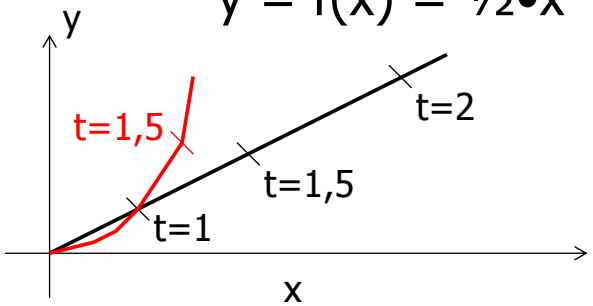


$X = 2 \bullet t^2$

$X = 2 \bullet t$



$y = t^2$

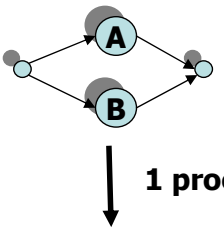


$y = f(x) = \frac{1}{2} \bullet x$

$y = \frac{1}{4} \bullet x^2$

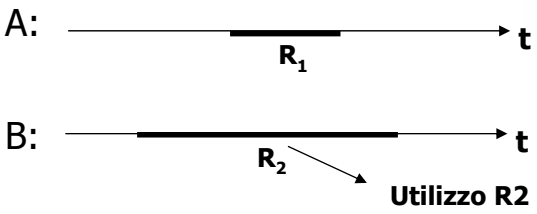
- nella rappresentazione 'mutua' y=f(x) il tempo è indicato sul percorso del moto
- non è regolare se la velocità non è costante

Evoluzione di processi

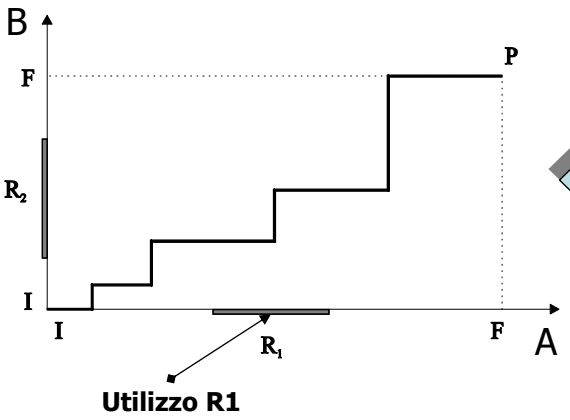


2 processori

1 processore



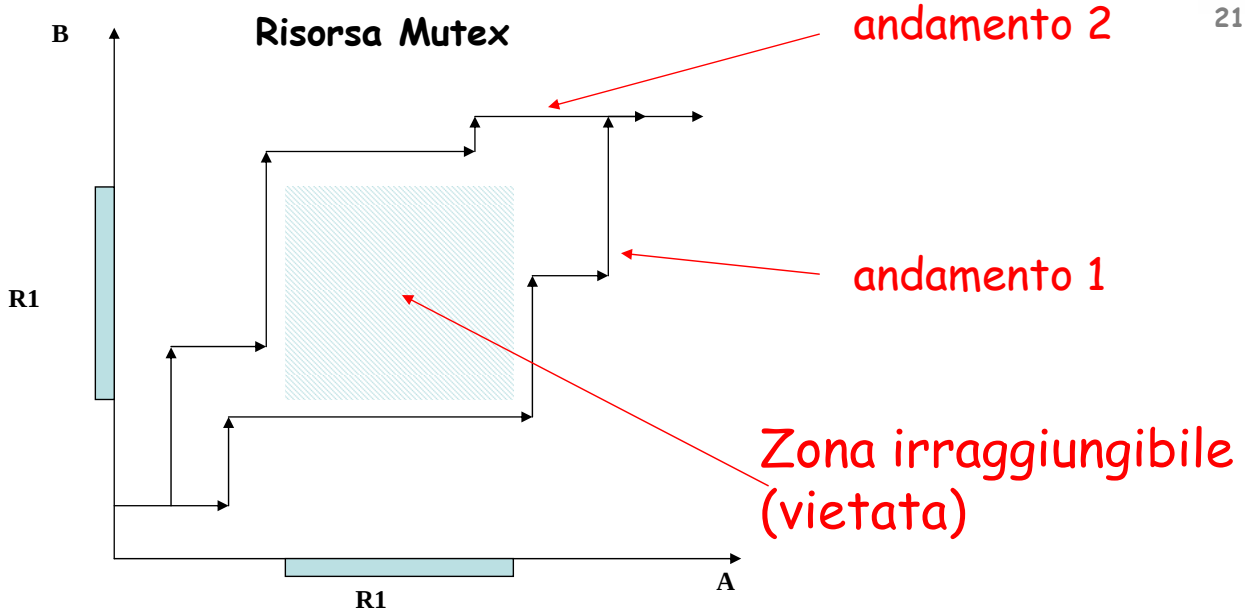
A usa la risorsa R_1
B usa la risorsa R_2



- Diagramma degli stati per 2 processi
- Non viene indicato il tempo
- È monotono non decrescente

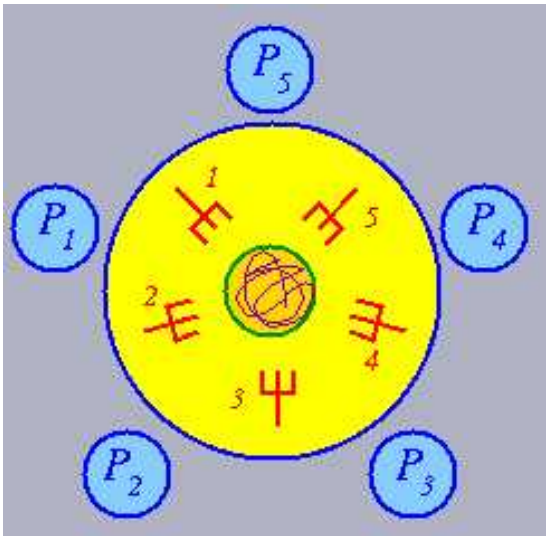
DEI UNIV PD © 2005

Mutua esclusione



DEI UNIV PD © 2005

Il problema dei 5 filosofi



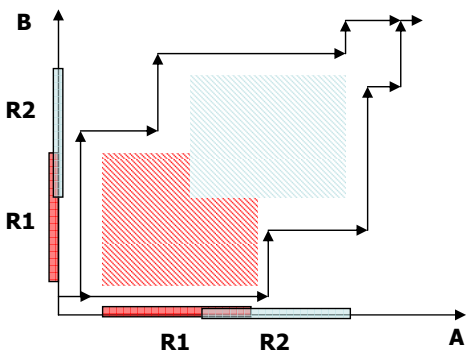
- I filosofi alternano fasi di riflessione a fasi di alimentazione
- Per mangiare devono dotarsi in modo esclusivo delle due forchette adiacenti
- È possibile arrivare ad una situazione di (deadlock) ovvero STALLO GLOBALE

DEI UNIV PD © 2005

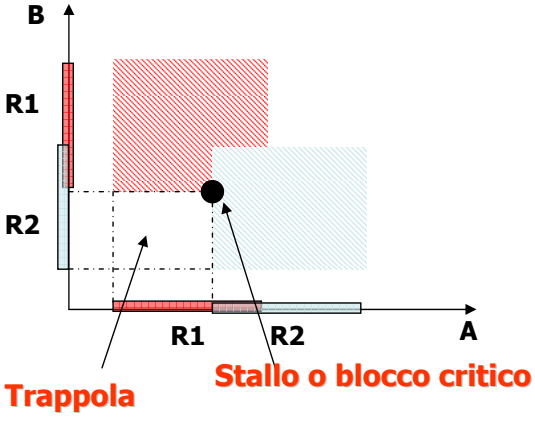
Stallo o Blocco Critico

- Un sistema di processi si dice in stallo se non può raggiungere il suo stato finale in quanto i processi si bloccano a vicenda.
- Si può verificare quando più processi usano risorse non condivisibili.
  Es. due processi richiedono R1 ed R2.
  Due possibilità:

Richiesta di risorse nello stesso ordine



Richiesta di risorse in ordine inverso



DEI UNIV PD © 2005

Condizioni di stallo

Sistema di processi in blocco critico

(condizioni necessarie)

Mutua esclusione: le risorse coinvolte non sono condivisibili.

&

Allocazione parziale: un processo non richiede tutte le risorse necessarie in un unico step, ma in diversi momenti della sua evoluzione.

&

Non sottraibilità: solo il processo che usa la risorsa è in grado di rilasciarla (no preemption).

&

Attesa circolare: i processi sono in attesa di risorse occupate da altri processi a loro volta in attesa di risorse occupate dai primi.

(ma non sufficienti)

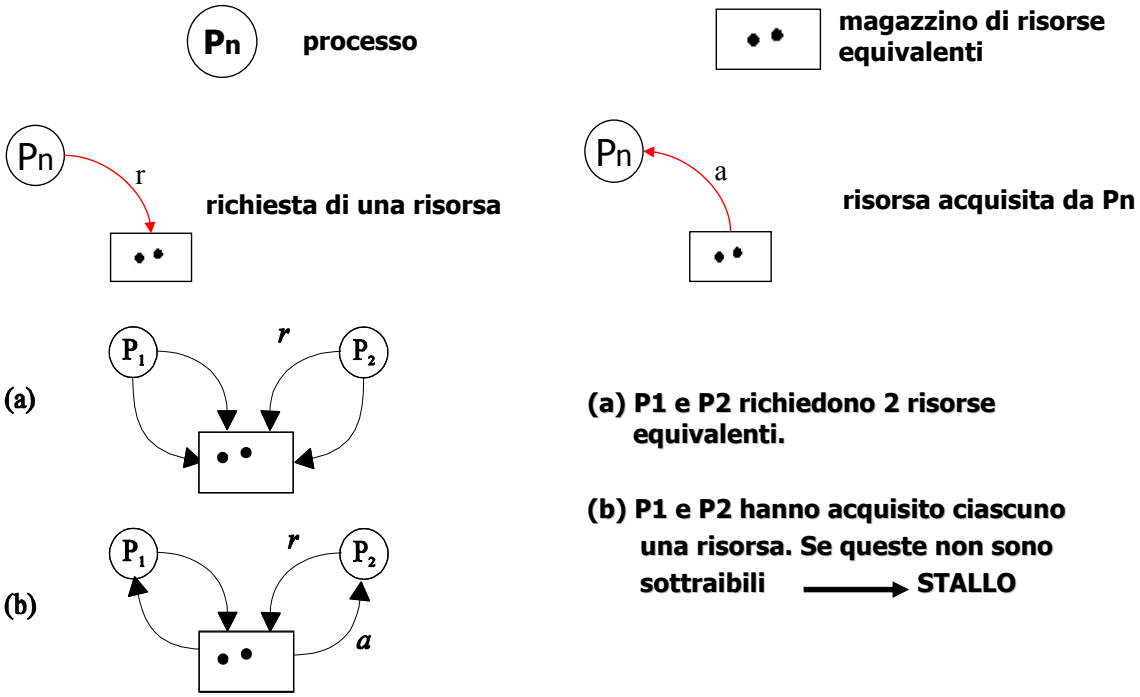
Soluzioni allo stallo

A posteriori: unica soluzione è la distruzione dei processi bloccati e di tutti quelli che da questi dipendono.

In effetti si tratta di una non-soluzione: non sempre è praticabile (es. risorse consumabili, sistemi real time).

A priori: politiche di allocazione delle risorse.

Algoritmo per individuare lo stallo



(a) P1 e P2 richiedono 2 risorse equivalenti.

(b) P1 e P2 hanno acquisito ciascuno una risorsa. Se queste non sono sottraibili ⟶ STALLO

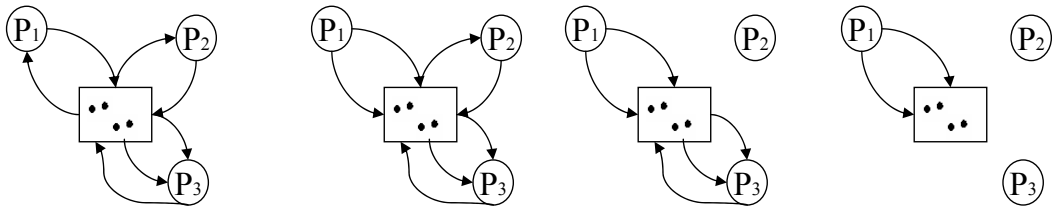
DEI UNIV PD © 2005

Algoritmo per individuare lo stallo [2]

Graficamente:

1. Definisco un ordinamento arbitrario dei processi i quali verranno esaminati nell'ordine prefissato.
2. Se un processo può acquisire tutte le risorse di cui abbisogna (se il residuo del magazzino è sufficiente a soddisfare il processo in questione), si è certi che esso termina e si possono eliminare i legami con il magazzino.
3. itero il ciclo.

Se non riesco ad eliminare tutti i legami, il sistema è in **stallo**.

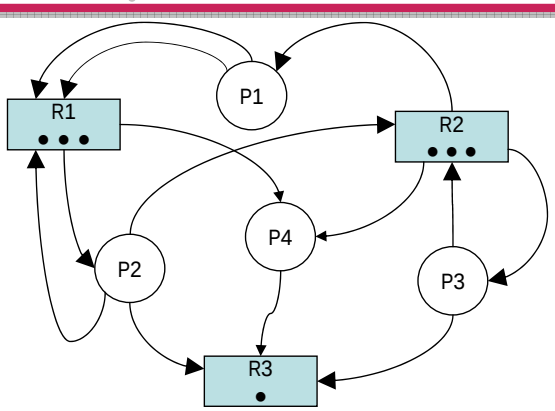


Sistema in stallo

Sistema non in stallo: applicazione dell'algoritmo

DEI UNIV PD © 2005

Esempio



DICHIARAZIONI

A	R_1	R_2	R_3
S	3	3	1
P_1	2	1	0
P_2	2	1	1
P_3	0	2	1
P_4	1	1	1

POSSESSO

B	R_1	R_2	R_3
S	2	3	0
P_1	0	1	0
P_2	1	0	0
P_3	0	1	0
P_4	1	1	0

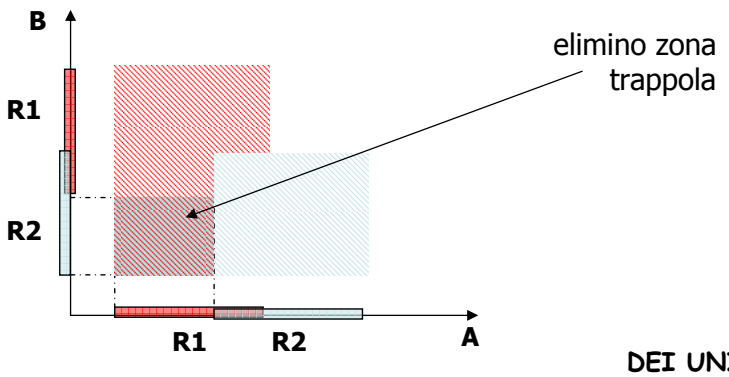
RICHIESTE

C	R_1	R_2	R_3
S	1	0	1
P_1	2	0	0
P_2	1	1	1
P_3	0	1	1
P_4	0	0	1

Prevenzione dello stallo

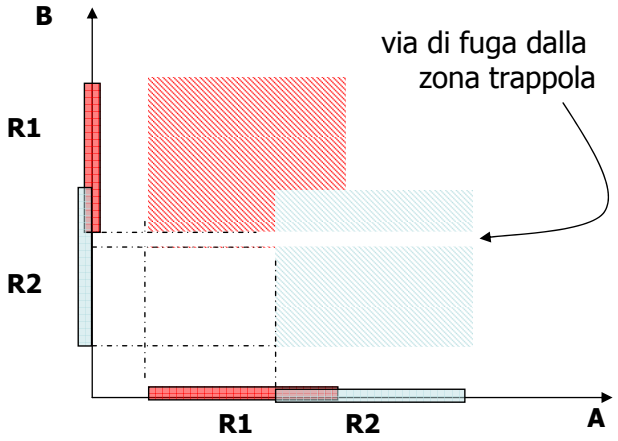
- Eliminare almeno una delle 4 condizioni necessarie per lo stallo.
  - Ciò può essere fatto mediante opportune politiche di allocazione delle risorse.

- Soluzione (A): allocazione globale delle risorse,
  - applicabile quando sono note le risorse utilizzate dai processi durante la loro evoluzione (rimuove la condizione dell'allocazione parziale).

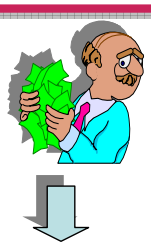


Prevenzione dello stallo [2]

- Soluzione (B): allocazione gerarchica delle risorse,
  - viene stabilito un ordinamento nelle risorse. Un processo che stia usando risorse, ne può ottenere altre solo di ordine più elevato.
  - Per ottenere risorse di ordine più basso, deve prima:
    - rilasciare quelle di ordine maggiore,
    - richiedere la risorsa di ordine inferiore,
    - richiedere di nuovo quelle superiori.



Ancora sulla prevenzione dello stallo



- Soluzione (C): Algoritmo del Banchiere, applicabile se si conosce il numero max di risorse richieste da un processo per terminare le sue attività.
- Le risorse vengono concesse ad un processo, solo se ci sono adeguate garanzie che esso giunga a termine rilasciando tutte le risorse acquisite.

Definiti:

D(0) = disponibilità di risorse nello stato iniziale; **D(0) > max(Mi)**
Mi = max n° di risorse per Pi;
Ai(s) = risorse allocate per Pi alla fase s;
Ri(s) = richiesta di risorse alla fase s.

In ogni fase è valida la seguente condizione:

Ai(s) <=Mi && Ai(s+1) = Ai(s)+Ri(s) <= Mi

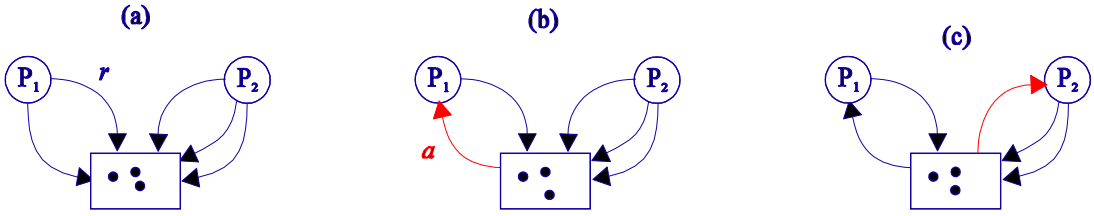
Si ha inoltre: **D(s)=D(0)-ΣAi(s)**. Allo stato s, le risorse richieste da Pi vengono concesse solo se il processo dà garanzie di terminare, ovvero è verificato **Mi-Ai(s) <= D(s)**.

Almeno un processo per ogni fase dà garanzie di terminare e, terminando, consente ad almeno un altro processo che ha ricevuto risorse in precedenza di terminare .... etc.

Algoritmo del banchiere

- È possibile rilassare la condizione di allocazione che impone la concessione delle risorse ad un processo, solo se sono immediatamente disponibili quelle che gli occorrono per terminare.

- È sufficiente che esista una sequenza di allocazione e rilascio nel sistema che,
  a partire dalla situazione attuale, garantisca la terminazione del sistema.



Nella fase c) la risorsa viene concessa a P2 in quanto esiste una sequenza di allocazione che porta tutti i processi a terminare.

Stallo individuale

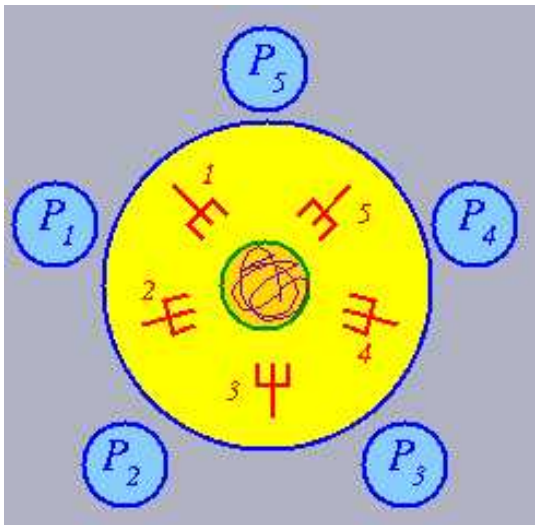
- Esiste un'altra forma di blocco che riguarda i singoli processi
  - Pur potendosi verificare la disponibilità delle risorse richieste da un processo, questo viene continuamente 'sopravanzato' da altri processi che chiedono meno risorse
  - La richiesta inferiore consente a questi di ottenere le risorse, per poi restituirle in una quantità che non è ancora sufficiente per soddisfare il primo processo

- Questa forma di blocco, che può prolungarsi per un tempo non definito, è detta **blocco individuale** o **starvation**

Esempio di starvation

- Pool di D0=10 risorse equivalenti, Alg. del Banchiere
- M(P1)=5 M(P2)=7 M(P3)=M(P4)=2
- R1(P1)=4 concesse, D1=6
- R2(P2)=7 non concesse, attende
- R3(P3)=2 concesse, D3=4
- R4(P4)=2 concesse, D4=2
- R5(P1)=1 concessa, D5=1
- P1 rilascia le risorse, D6=6
- R7(P1)=4 concesse, D7=2
- con questo tipo di sequenza P2 attende per un tempo non definito

Starvation nei 5 filosofi



- Sequenza di starvation:

  P1 mangia
  P3 mangia
  (nessun altro può mangiare)
  P3 pensa
  [P4 può mangiare]
  P3 mangia
  P1 pensa
  [P5 può mangiare]
  P1 mangia

- P2 subisce starvation

Fine

02.b

La concorrenza e le risorse