



deadlock

capitolo 7 del libro (VII ed.)

Introduzione

- **Deadlock:** situazione per cui un insieme di processi sono fermi in attesa di un evento che solo uno dei processi appartenenti all'insieme stesso potrebbe causare
- Noi vedremo il problema del deadlock in relazione al problema della gestione di risorse, anche se si tratta di un **problema più generale**

Introduzione

- **Deadlock:** situazione per cui un insieme di processi sono fermi in attesa di un evento che solo uno dei processi appartenenti all'insieme stesso potrebbe causare
- **Problema della gestione delle risorse:**
 - un sistema fisico può essere visto come un insieme di **risorse**
 - ogni risorsa può essere presente in un certo numero di **istanze equivalenti**, es:
 - CPU: 2 (sistema biprocessore)
 - stampanti: 3
 - CD-reader: 1

Introduzione

- Un processo che vuole usare N istanze di una certa risorsa deve farne richiesta al gestore delle risorse, cioè al SO
- Si distinguono 3 fasi:

- **richiesta:** può causare attesa ●
- uso
- **rilascio** ●



sono effettuati tramite system call

-
- Il SO tiene traccia di quali risorse sono assegnate a quale processo
 - Richiesta/rilascio: tramite **system call ad hoc** (es. open e close di file) oppure tramite **strumenti di sincronizzazione** (es. semafori)

Condizioni al deadlock

- Es. di deadlock fra due processi: P1 detiene l'unico lettore di fotografie e vuole la stampante per stampare delle foto, P2 ha la stampante e vuole il lettore perché deve, anch'esso, stampare delle foto ...
- **Condizioni necessarie al deadlock**

<1> **ME**: risorse non condivisibili

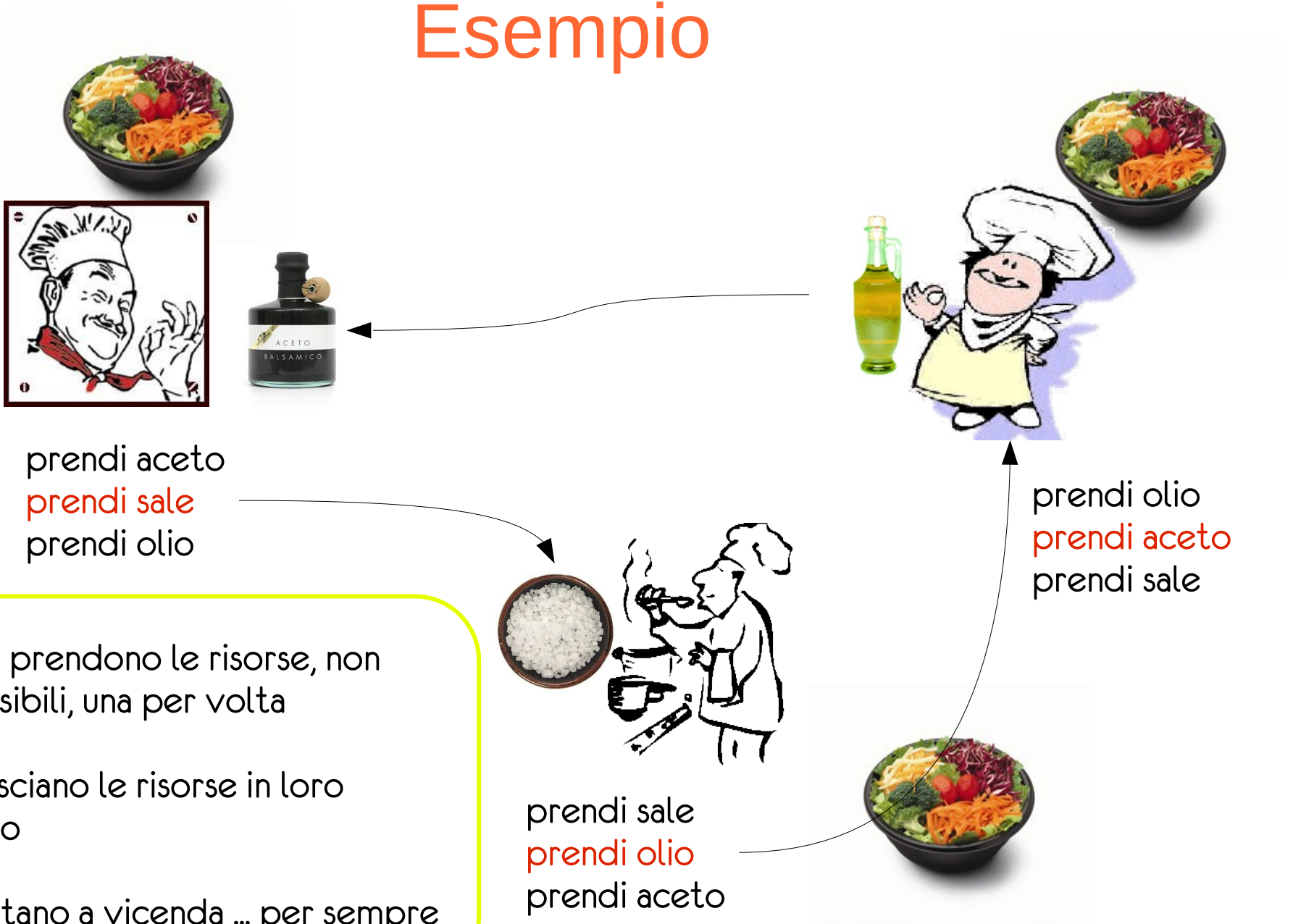
<2> **possesso e attesa**: un processo attende le risorse non disponibili, anche detenendo già il possesso di alcune delle risorse a lui necessarie

<3> **no prelazione**: il rilascio non viene forzato

<4> **attesa circolare**: siano i processi in questione P_1, \dots, P_n , allora P_1 attende risorse da P_2 , che attende risorse da P_3 , ecc. e P_n attende risorse da P_1

basta che una **non** sia vero per evitare il deadlock

Esempio



i cuochi prendono le risorse, non
condivisibili, una per volta

non rilasciano le risorse in loro
possesso

si aspettano a vicenda ... per sempre

In codice

Cuoco1

(1) P(olio);
(2) P(aceto);
(3) P(sale);

... cucina ...

rilascia risorse

Cuoco2

(1) P(sale);
(2) P(olio);
(3) P(aceto);

... cucina ...

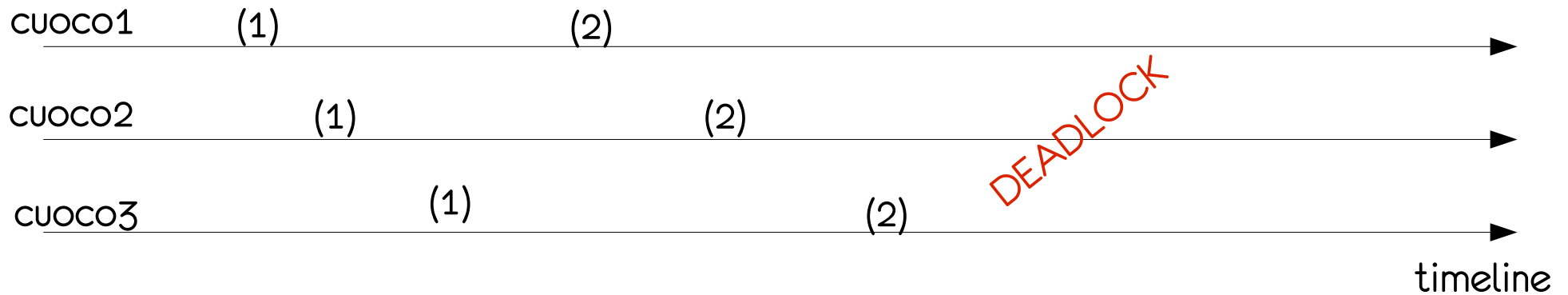
rilascia risorse

Cuoco3

(1) P(aceto);
(2) P(sale);
(3) P(olio);

... cucina ...

rilascia risorse



In codice

Cuoco1

(1) P(olio);
(2) P(aceto);
(3) P(sale);

... cucina ...

rilascia risorse

Cuoco2

(1) P(sale);
(2) P(olio);
(3) P(aceto);

... cucina ...

rilascia risorse

Cuoco3

(1) P(aceto);
(2) P(sale);
(3) P(olio);

... cucina ...

rilascia risorse

