

Sistemi di elaborazione in tempo reale: un'introduzione

Eugenio Faldella

Dipartimento di Informatica - Scienza e Ingegneria
Scuola di Ingegneria e Architettura, Università di Bologna



eugenio.faldella@unibo.it
<http://www.ing.unibo.it>

SISTEMI DI ELABORAZIONE IN TEMPO REALE

Un sistema di elaborazione opera in tempo reale soltanto se fornisce i risultati attesi entro prestabiliti limiti temporali (dipendenti dal contesto applicativo).

Quali gli obiettivi di progetto (del software) ?

non solo

- ◆ correttezza
- ◆ efficienza (!?)
- ◆ affidabilità
- ◆ flessibilità
- ◆ portabilità
- ◆ riusabilità

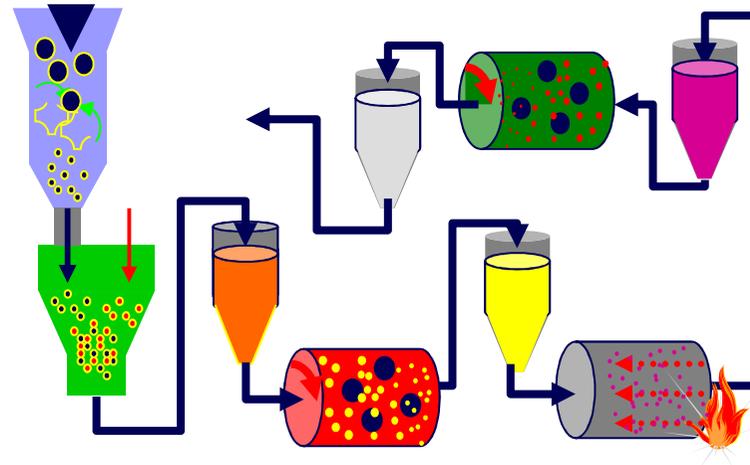
ma anche

- ◆ predicibilità essenziale

UN TIPICO (regionale) CONTESTO APPLICATIVO

controllo di processi
e automazione industriale

Emilia Romagna:
"Packaging Valley"



C95:
11.6 blister/s



G.D

H1000:
1.6 stecche/s

aspetti temporali

aspetti funzionali

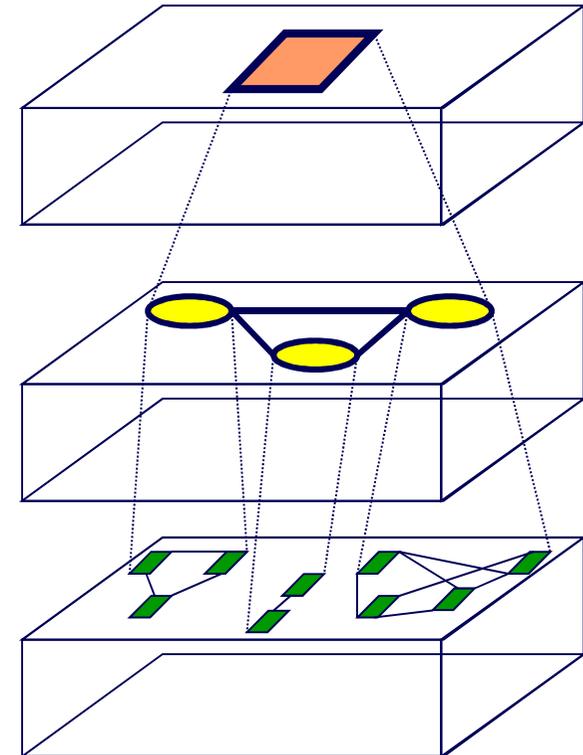
ASPETTI FUNZIONALI: METODOLOGIA DI PROGETTO "TOP-DOWN"

Livelli di descrizione

La descrizione del comportamento di un sistema complesso, in ogni ambito ingegneristico, è inevitabilmente articolata su più livelli (approccio "divide et impera").

Ogni livello individua entità astratte o concrete opportunamente cooperanti, contraddistinte da ben definiti ruoli, funzionalità, interfacce e protocolli di interazione con le altre entità operanti nello stesso livello o nei livelli adiacenti della gerarchia.

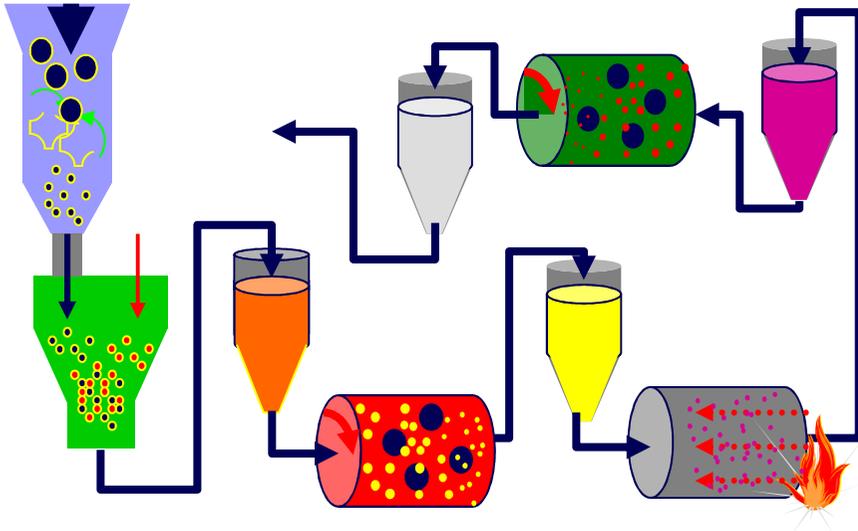
Esplorando i livelli della gerarchia dall'alto verso il basso, aumenta il numero di componenti ma diminuisce la complessità di ciascuno di essi, dal punto di vista sia comportamentale che strutturale.



How should decomposition be carried out ? The answer lies at the heart of all software design activities. [1]

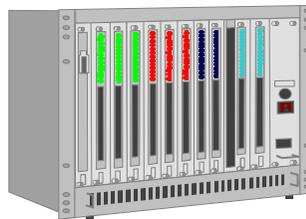
The best software designs look simple, but it takes a lot of hard work to design a simple architecture. [2]

ASPETTI TEMPORALI: UN PROBLEMA FONDAMENTALE



- selezionata l'architettura del sistema di elaborazione

- ◆ monoprocesore
- ◆ multiprocesore



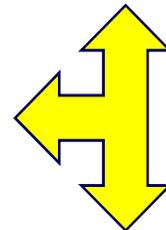
- definita un'applicazione in termini di processi cooperanti caratterizzati da:

- ◆ prefissate interazioni

- vincoli di precedenza
- risorse condivise

- ◆ prefissate specifiche temporali

- frequenza (max) di esecuzione
- tempo max di elaborazione richiesto ad ogni esecuzione
- tempo limite di completamento di ogni elaborazione



occorre individuare un'opportuna strategia di schedulazione dei processi in modo tale da rispettare tutti i vincoli imposti dall'applicazione

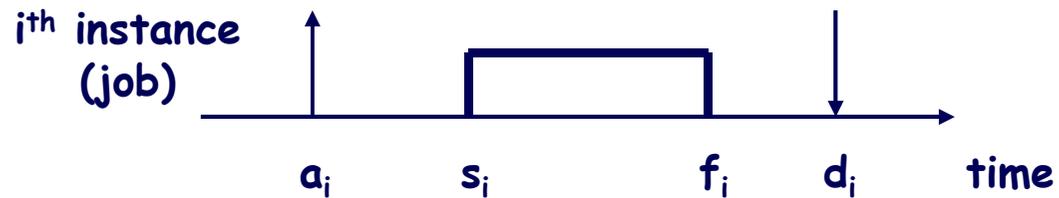
TIPOLOGIE DI SCHEDULAZIONE

- off-line se integralmente pianificata a priori
- on-line se stabilita a tempo di esecuzione
 in base a parametri attribuiti ai processi in maniera
 - ◆ static
 - ◆ dynamic
- guaranteed se rispetta i vincoli temporali di tutti i processi
- best-effort se tende viceversa ad ottimizzare le prestazioni medie dell'insieme di processi
- preemptive se l'esecuzione di un processo può essere sospesa
- non-preemptive in caso contrario

TIPOLOGIE DI PROCESSO

- **real-time** con vincoli temporali
 - **hard** se i vincoli temporali devono sempre essere rispettati
 - ◆ **periodico** con frequenza di esecuzione costante
 - ◆ **sporadico** in caso contrario
 - **soft** se i vincoli temporali possono essere disattesi in condizioni di temporaneo sovraccarico
 - ◆ **periodico** con frequenza di esecuzione costante
 - ◆ **aperiodico** in caso contrario
- **non real-time** senza vincoli temporali

PARAMETRI TEMPORALI DI UN PROCESSO



- a_i (r_i) arrival (release) time
- d_i deadline
- s_i start time
- f_i finishing time

- $C_i = f_i - s_i$ computation time
- $D_i = d_i - a_i$ relative deadline
- $R_i = f_i - a_i$ response time
- $L_i = f_i - d_i$ lateness
- $E_i = \max(0, L_i)$ tardiness (exceeding time)
- $X_i = D_i - C_i$ laxity (slack time)

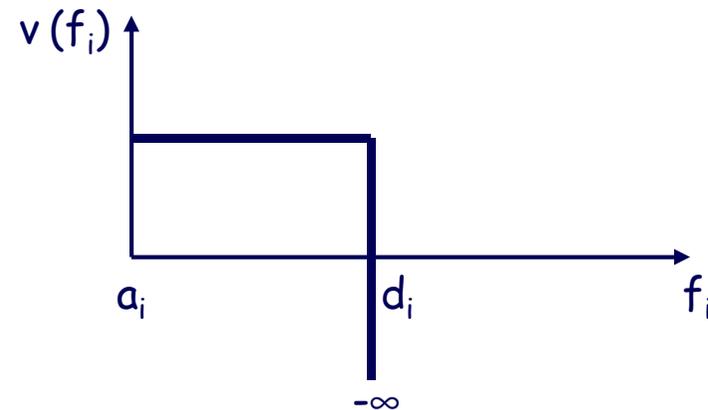
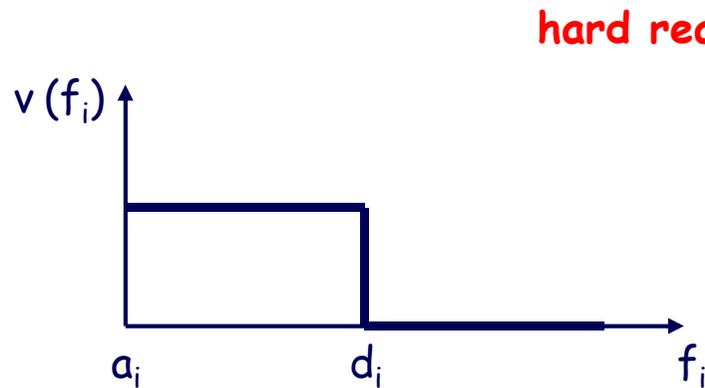
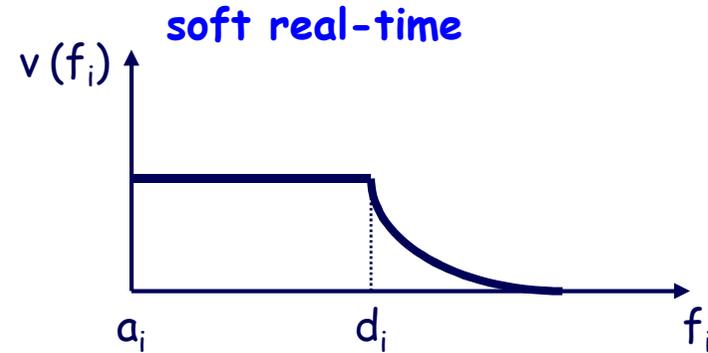
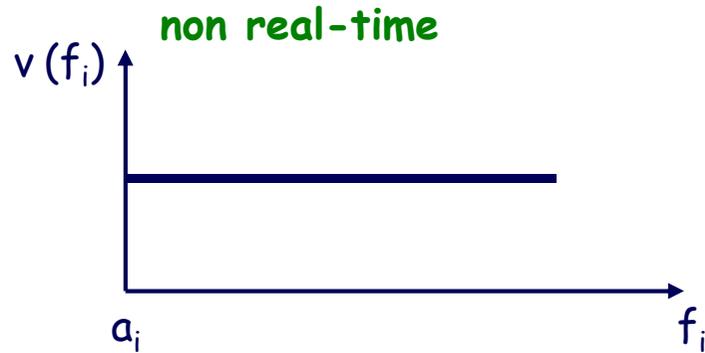
Processi periodici:

- $a_{i+1} - a_i = T$ (period)
- $D_i = T$
- $a_1 = \phi$ (phase)

Processi sporadici:

- $a_{i+1} - a_i \geq MIT$ (minimum interarrival time)
- $D_i < MIT$

FUNZIONE DI UTILITA' DI UN PROCESSO



“better never than late”

TEMPO REALE NON IMPLICA NECESSARIAMENTE [GRAHAM 76] ...

Sia A un'applicazione costituita da 9 processi $P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9$,

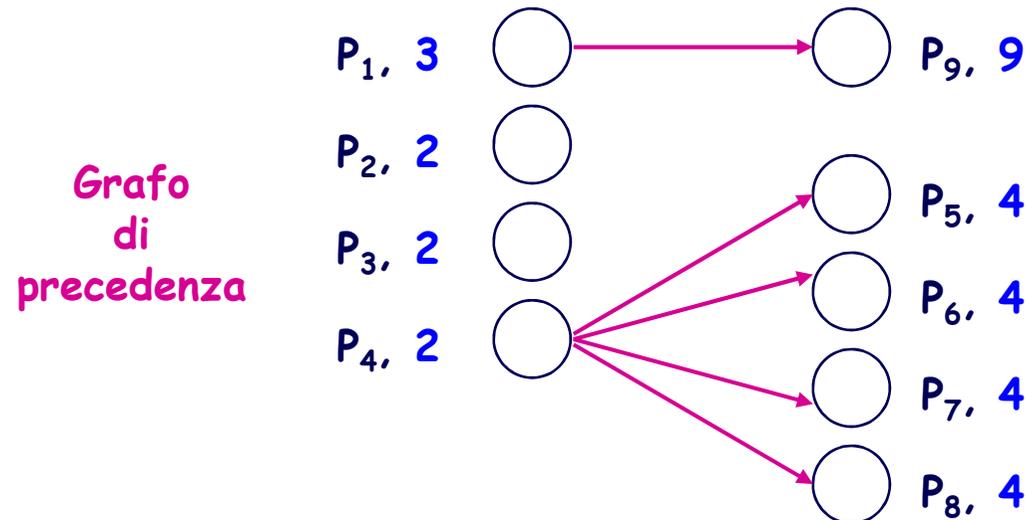
di priorità decrescente $p_1 > p_2 > p_3 > p_4 > p_5 > p_6 > p_7 > p_8 > p_9$

con i seguenti vincoli di precedenza ($\prec \equiv$ precede)

$P_1 \prec P_9, P_4 \prec P_5, P_4 \prec P_6, P_4 \prec P_7, P_4 \prec P_8$

e caratterizzati dai seguenti tempi di esecuzione [t.u.]

$C_1 = 3, C_2 = C_3 = C_4 = 2, C_5 = C_6 = C_7 = C_8 = 4, C_9 = 9$



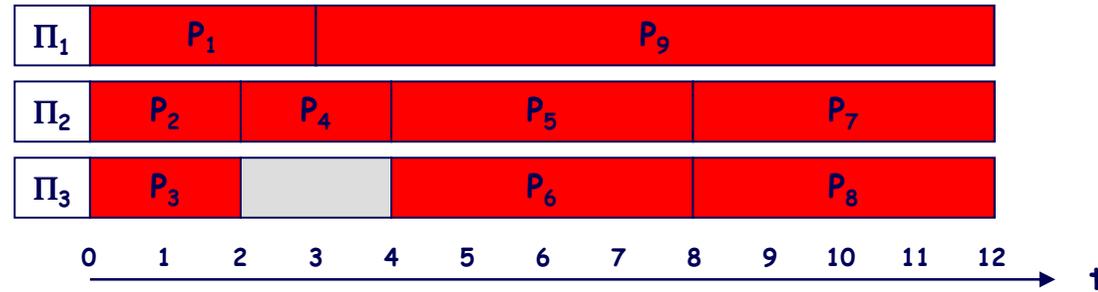
Sia E un sistema di elaborazione comprendente k processori

$\Pi_1 \Pi_2 \dots \Pi_k$

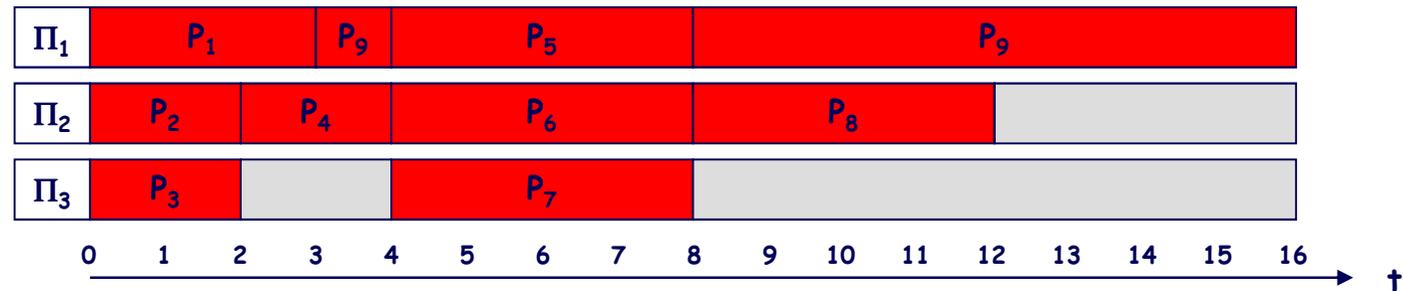
PREEMPTIVE SCHEDULING

$k = 3$

non-preemptive scheduling



preemptive scheduling



tempo di risposta: + 33.3 %



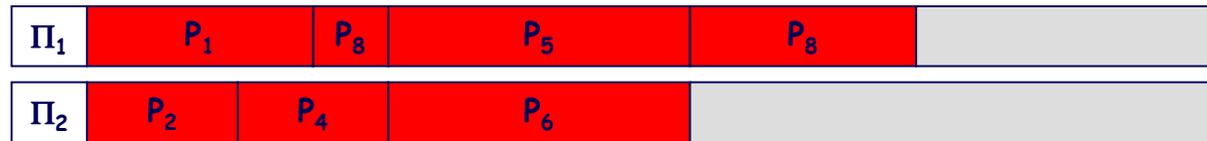
RILASSAMENTO DEI VINCOLI DI PRECEDENZA

$P_4 \text{ non } < P_7, P_8$

non-preemptive scheduling



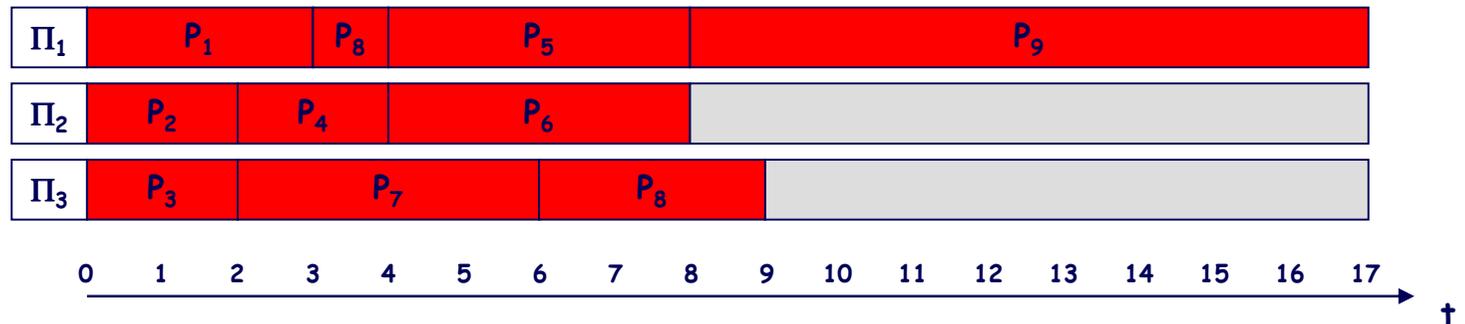
preemptive scheduling



multi-uniprocessor



symmetric multiprocessor



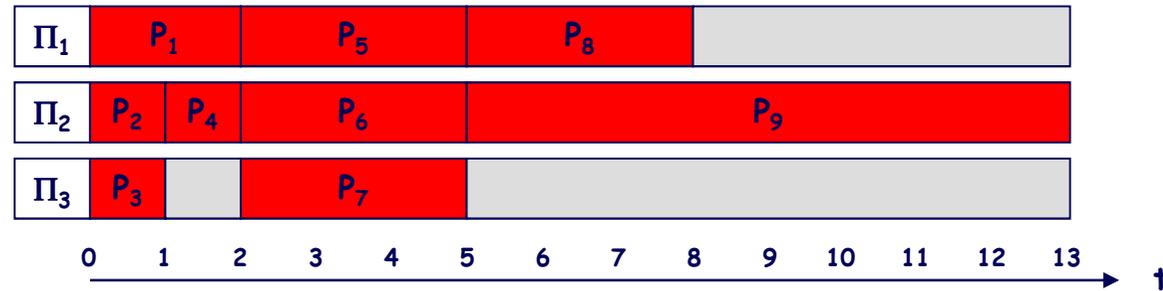
tempo di risposta: + 33.3 % / + 25 % / + 41.6 %



PROGRAMMAZIONE AD HOC

$$C_j - 1, \forall j$$

preemptive/non-preemptive scheduling



tempo di risposta: + 8.3 % ☹️

ELEVATA CAPACITÀ COMPUTAZIONALE

k = 4

preemptive/non-preemptive
scheduling



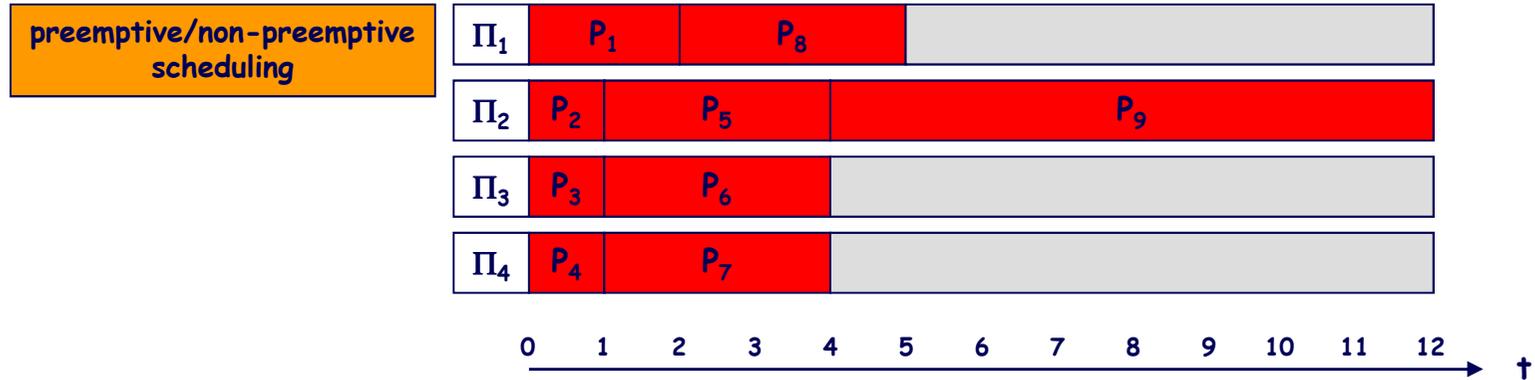
tempo di risposta: + 25 % ☹️

O TUTTO CIO' CONGIUNTAMENTE

$P_4 \text{ non } < P_7, P_8$

$C_j - 1, \forall j$

$k = 4$



tempo di risposta: - 0 % ☹️

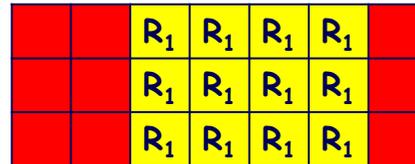


SCHEDULAZIONE DI PROCESSI IN PRESENZA DI RISORSE CONDIVISE ...

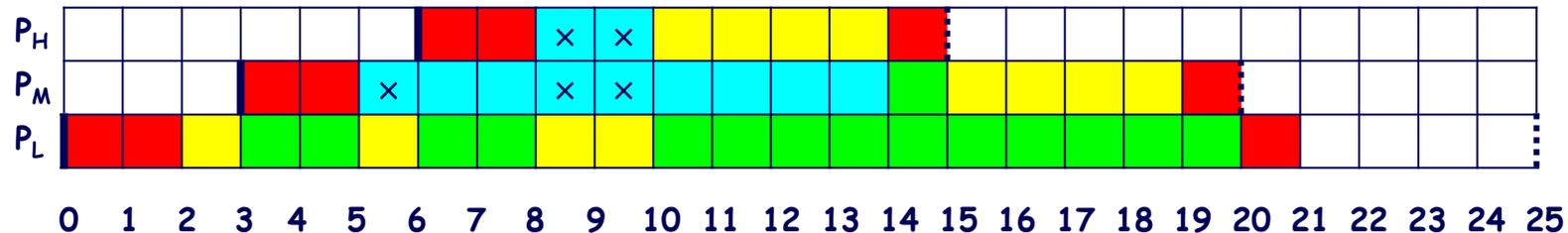
Quali problemi ?

A_1 : tre processi P_H , P_M , P_L , contraddistinti dai seguenti parametri temporali, condividono una stessa risorsa (R_1).
L'accesso mutuamente esclusivo alla risorsa è regolamentato da un semaforo (S_1).

	priorità	a [t.u.]	d [t.u.]	C [t.u.]
P_H	p_H (max)	6	15	7
P_M	p_M	3	20	7
P_L	p_L (min)	0	25	7



All'attivazione ogni processo prevede:
normale esecuzione (2 t.u.)
lock (S_1)
esecuzione della sezione critica (4 t.u.)
unlock (S_1)
normale esecuzione (1 t.u.)



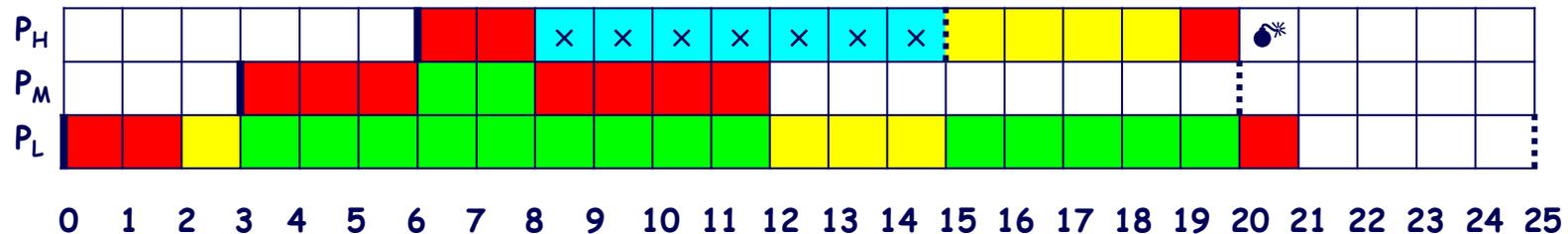
Il blocco (x) che P_H (nell'intervallo (8-10]) e P_M (negli intervalli (5-6] e (8-10]) subiscono da parte di P_L è inevitabile: accesso mutuamente esclusivo alla risorsa condivisa R_1 .

... PROCESSI CON RISORSE CONDIVISE ...

Il problema dell'inversione di priorità incontrollata

A_2 : tre processi P_H , P_M , P_L , contraddistinti dai seguenti parametri temporali, ed una risorsa R_1 condivisa da P_H e P_L .

	priorità	a [t.u.]	d [t.u.]	C [t.u.]	
P_H	p_H (max)	6	15	7	
P_M	p_M	3	20	7	
P_L	p_L (min)	0	25	7	



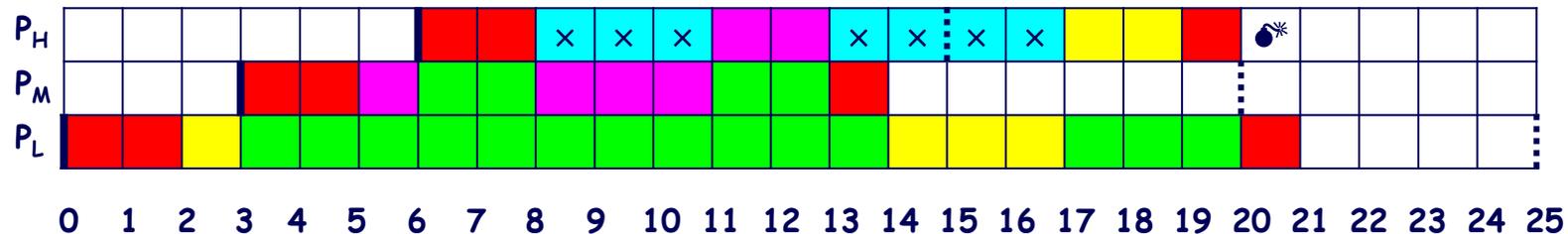
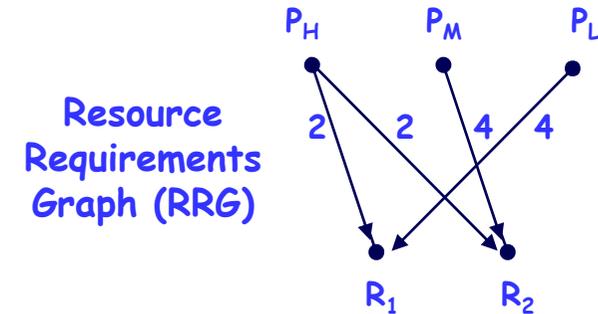
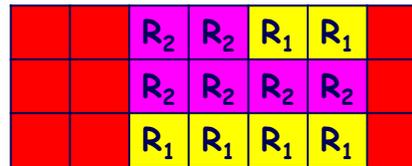
Il blocco che P_H subisce nell'intervallo (8-12] da parte di P_M è assolutamente da evitare.

... PROCESSI CON RISORSE CONDIVISE ...

Il problema della concatenazione dei blocchi

A_3 : tre processi P_H , P_M , P_L , contraddistinti dai seguenti parametri temporali, condividono due risorse R_1 , R_2 .

	priorità	a [t.u.]	d [t.u.]	C [t.u.]
P_H	p_H (max)	6	15	7
P_M	p_M	3	20	7
P_L	p_L (min)	0	25	7



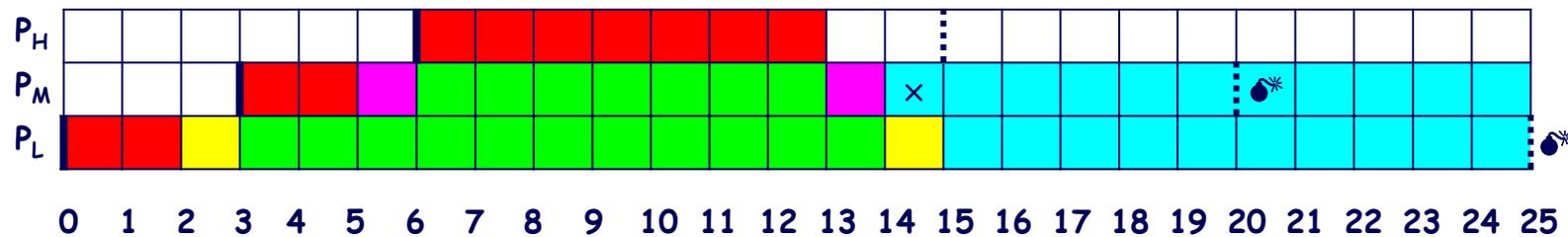
La concatenazione dei blocchi che P_H subisce negli intervalli (8-11] e (14-17] è evitabile (è peraltro da evitare l'inversione di priorità che si verifica nell'intervallo (13-14]).

... PROCESSI CON RISORSE CONDIVISE

Il problema del deadlock

A₄: tre processi P_H, P_M, P_L, contraddistinti dai seguenti parametri temporali, e due risorse R₁, R₂ condivise da P_M e P_L.

	priorità	a [t.u.]	d [t.u.]	C [t.u.]	
P _H	p _H (max)	6	15	7	
P _M	p _M	3	20	7	
P _L	p _L (min)	0	25	7	



Problema di minor rilievo, dal momento che le situazioni di deadlock possono essere a priori evitate rispettando opportune regole di ordinamento degli accessi annidati a risorse condivise.

PROTOCOLLI DI ACCESSO A RISORSE CONDIVISE

In presenza di risorse condivise con accessi regolamentati da classici meccanismi semaforici, un processo può essere bloccato, anche più volte, da processi di priorità inferiore. Un blocco può essere:

- **inevitabile** (accesso mutuamente esclusivo)
- **evitabile** (concatenazione di blocchi)
- **da evitare** (inversione di priorità incontrollata)

I tempi di blocco che conseguentemente possono subire i processi risultano:

- **limitati ma potenzialmente significativi** se si verifica una concatenazione di blocchi;
- **illimitati** se si verifica un'inversione di priorità incontrollata;
- **infiniti** se si verifica una situazione di deadlock.

Al fine di garantire la schedulabilità di un'applicazione occorre limitare il massimo tempo di blocco dei processi, avvalendosi allo scopo di efficaci "*protocolli di accesso a risorse condivise*", ovvero di insiemi di regole che definiscono:

- in quali condizioni consentire ai processi l'accesso alle risorse;
- come gestire le priorità dei processi allorché detengono risorse;
- come operare la schedulazione dei processi.