

# I sistemi di elaborazione in tempo reale: caratteristiche generali, peculiarità e problematiche di progetto

Eugenio Faldella

Dipartimento di Elettronica, Informatica e Sistemistica

Facoltà di Ingegneria

Università di Bologna

# 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) ?

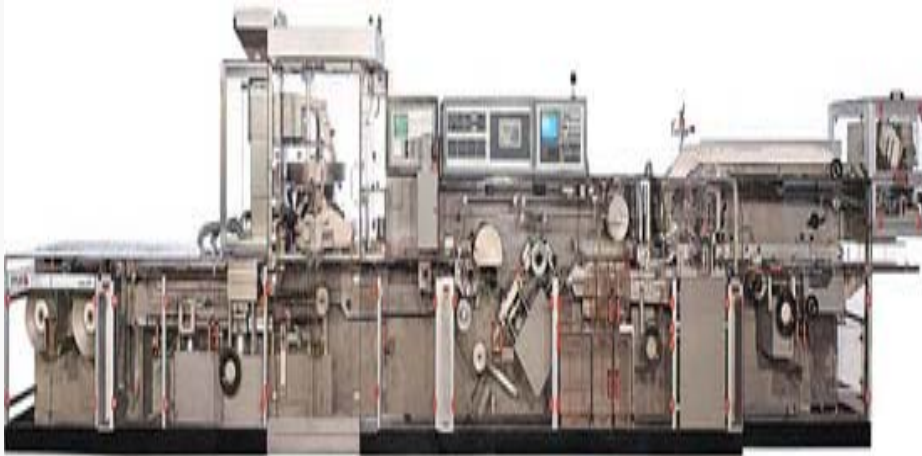
non solo

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

ma anche

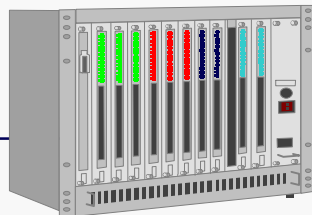
- ◆ predicibilità essenziale

# UN PROBLEMA FONDAMENTALE



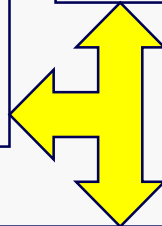
- selezionata l'architettura del sistema di elaborazione

- ◆ monoprocessore
- ◆ multiprocessore



- 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 esecuzione dei processi in modo tale da rispettare tutti i vincoli imposti dall'applicazione

# TIPOLOGIE DI SCHEDULAZIONE

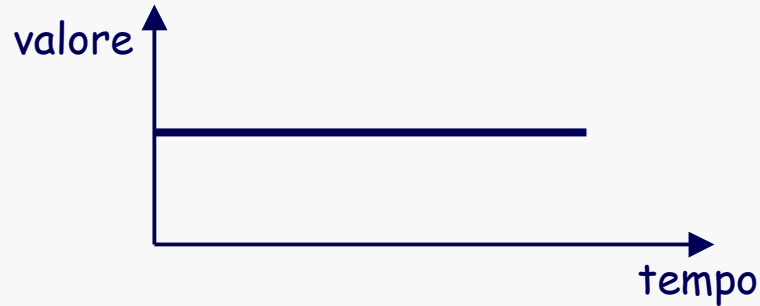
- off-line se integralmente pianificata a priori
- on-line se stabilita a tempo di esecuzione  
in base a parametri attribuiti ai processi in maniera
  - ◆ static
  - ◆ dynamic
- guaranteed se rispetta i vincoli temporali di tutti i processi
- best-effort se tende viceversa ad ottimizzare le prestazioni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

# TIPOLOGIE DI PROCESSO

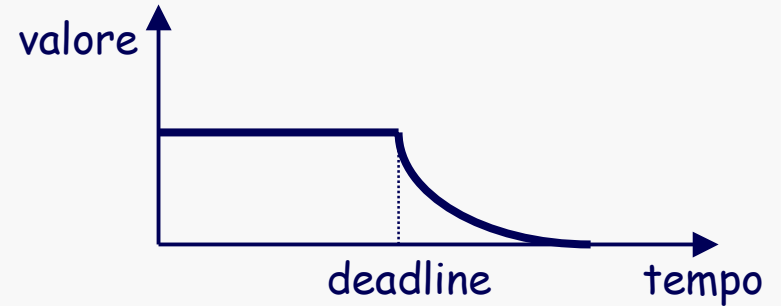
- ▼ **real-time**      con vincoli temporali
  - **hard**      se la relativa deadline deve sempre essere rispettata
    - ◆ **periodico**      con frequenza di esecuzione costante
    - ◆ **sporadico**      in caso contrario
  - **soft**      se la relativa deadline può essere disattesa in condizioni di temporaneo sovraccarico
    - ◆ **periodico**      con frequenza di esecuzione costante
    - ◆ **aperiodico**      in caso contrario
- ▼ **non real-time**      senza vincoli temporali

# FUNZIONE DI UTILITA' DI UN PROCESSO

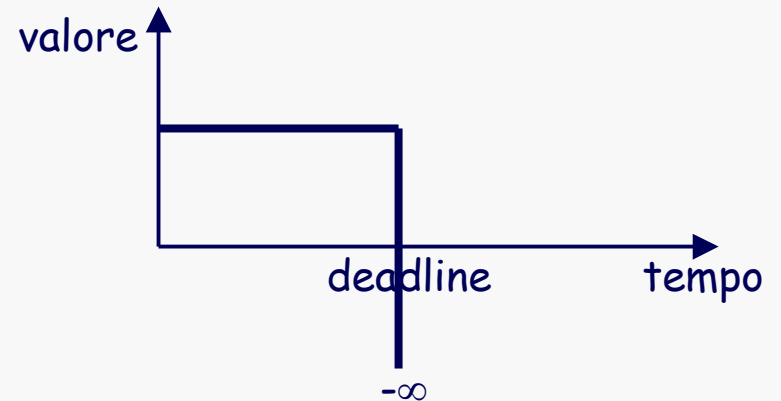
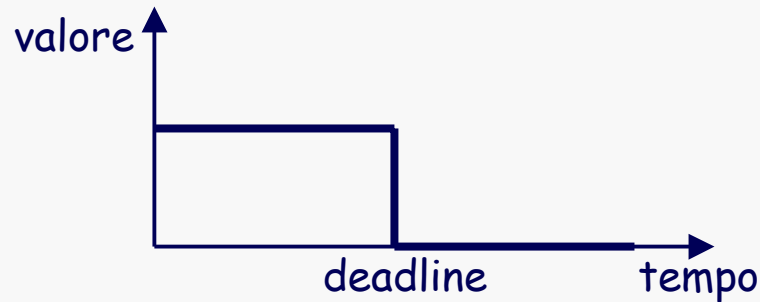
non real-time



soft real-time



hard real-time



"better never than late"

# TEMPO REALE NON SIGNIFICA NECESSARIAMENTE ...

Sia  $A_1$  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

$p(P_i) < p(P_j) \forall i, j$  con  $i > j$

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

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

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

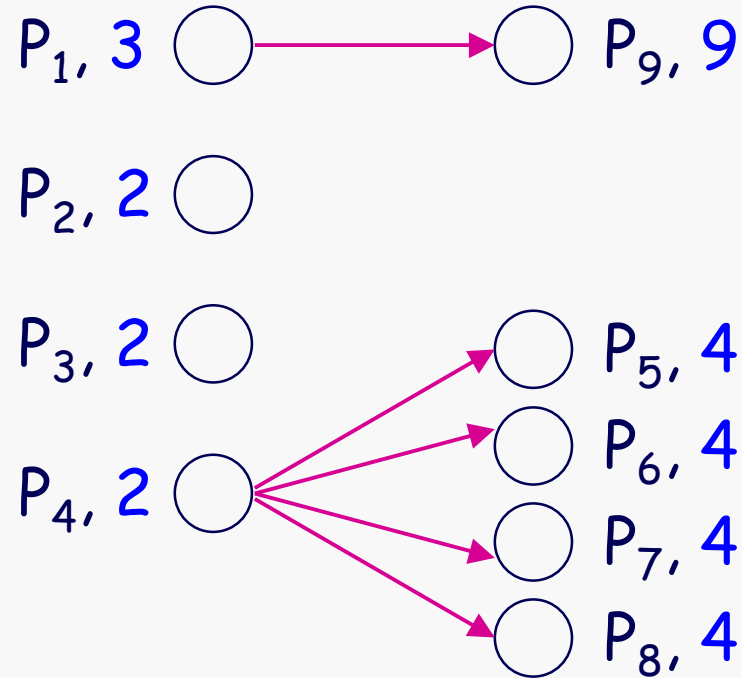
$C_1 = 3, C_2 = C_3 = C_4 = 2, C_5 = C_6 = C_7 = C_8 = 4, C_9 = 9$

Sia  $E$  un sistema di elaborazione comprendente  $k$  processori

$\Pi_1 \Pi_2 \dots \Pi_k$

# ... TEMPO REALE NON SIGNIFICA NECESSARIAMENTE ...

Grafo  
di  
precedenza



$k = 3$

$\Pi_1$	$P_1$	$P_9$		
$\Pi_2$	$P_2$	$P_4$	$P_5$	$P_7$
$\Pi_3$	$P_3$		$P_6$	$P_8$



# ... PREEMPTIVE SCHEDULING

no preemption → preemption

$\Pi_1$	$P_1$	$P_9$		
$\Pi_2$	$P_2$	$P_4$	$P_5$	$P_7$
$\Pi_3$	$P_3$		$P_6$	$P_8$

$\Pi_1$	$P_1$	$P_9$	$P_5$	$P_9$	
$\Pi_2$	$P_2$	$P_4$	$P_6$	$P_8$	
$\Pi_3$	$P_3$		$P_7$		



preemptive scheduling → tempo di risposta ↑

☹️ +33.3%

# ... PROGRAMMAZIONE AD HOC

$C_j \rightarrow C_j - 1 \quad \forall j$

$\Pi_1$	$P_1$		$P_9$		
$\Pi_2$	$P_2$	$P_4$	$P_5$		$P_7$
$\Pi_3$	$P_3$		$P_6$		$P_8$

$\Pi_1$	$P_1$		$P_5$		$P_8$		
$\Pi_2$	$P_2$	$P_4$	$P_6$		$P_9$		
$\Pi_3$	$P_3$		$P_7$				



↓ carico computazionale → tempo di risposta ↑

 +8.3%

# ... ELEVATA CAPACITÀ COMPUTAZIONALE

$k = 3 \rightarrow k = 4$

$\Pi_1$	$P_1$	$P_9$		
$\Pi_2$	$P_2$	$P_4$	$P_5$	$P_7$
$\Pi_3$	$P_3$		$P_6$	$P_8$

$\Pi_1$	$P_1$	$P_8$		
$\Pi_2$	$P_2$	$P_5$	$P_9$	
$\Pi_3$	$P_3$	$P_6$		
$\Pi_4$	$P_4$	$P_7$		



↑ capacità computazionale → tempo di risposta ↑

 +25%

# UN PRIMO MODELLO DI RIFERIMENTO PER LA SCHEDULAZIONE DI PROCESSI HARD REAL-TIME (PERIODICI E/O SPORADICI) ...

- N processi  $P_1, P_2, \dots, P_N$  indipendenti
  - ◆ senza vincoli di precedenza
  - ◆ senza risorse condivise
- ogni processo  $P_j$  ( $j = 1, 2, \dots, N$ ) è caratterizzato da tre parametri temporali
  - ◆  $T_j$ : periodo nel caso di processi periodici, minimum interarrival time nel caso di processi sporadici
  - ◆  $D_j$ : deadline relativa, con  $D_j = T_j$  nel caso di processi periodici,  $D_j < T_j$  nel caso di processi sporadici
  - ◆  $C_j$ : tempo massimo di esecuzione, con  $C_j \leq D_j$
- l'esecuzione dei processi è affidata ad un sistema di elaborazione monoprocesso

# ... UN PRIMO MODELLO DI RIFERIMENTO PER LA SCHEDULAZIONE DI PROCESSI HARD REAL-TIME (PERIODICI E/O SPORADICI)

- ad ogni processo  $P_j$  è staticamente o dinamicamente associata una priorità  $p(P_j) = p_j$  dipendente dalla corrispondente deadline, relativa o assoluta rispettivamente
- ad ogni processo è dinamicamente associata una informazione che ne identifica lo stato ai fini della schedulazione:
  - ◆ processo IDLE, READY (SUSPENDED), RUNNING
- l'esecuzione di un processo è sospesa quando un altro processo di priorità superiore è pronto per l'esecuzione (preemption)
- il tempo impiegato dal processore per operare una commutazione di contesto tra processi è trascurabile

## LA STRATEGIA RATE MONOTONIC PRIORITY ORDERING (RMPO)

Tipologia di processi: periodici.

Ad ogni processo è staticamente associata una priorità tanto maggiore quanto minore è la corrispondente deadline relativa (periodo).

## LA STRATEGIA DEADLINE MONOTONIC PRIORITY ORDERING (DMPO)

Tipologia di processi: periodici e/o sporadici.

Ad ogni processo è staticamente associata una priorità tanto maggiore quanto minore è la corrispondente deadline relativa.

## LA STRATEGIA EARLIEST DEADLINE FIRST (EDF)

Tipologia di processi: periodici e/o sporadici.

Ad ogni processo è dinamicamente associata una priorità tanto maggiore quanto più prossima è la corrispondente deadline assoluta.

# UN TEOREMA SULLA SCHEDULABILITÀ

I requisiti temporali dei processi sono coerenti e consistenti ?

Condizione necessaria (ma in generale non sufficiente)  
affinché un insieme di N processi periodici sia schedulabile  
è che il risultante fattore di utilizzazione del processore  
sia non superiore a 1:

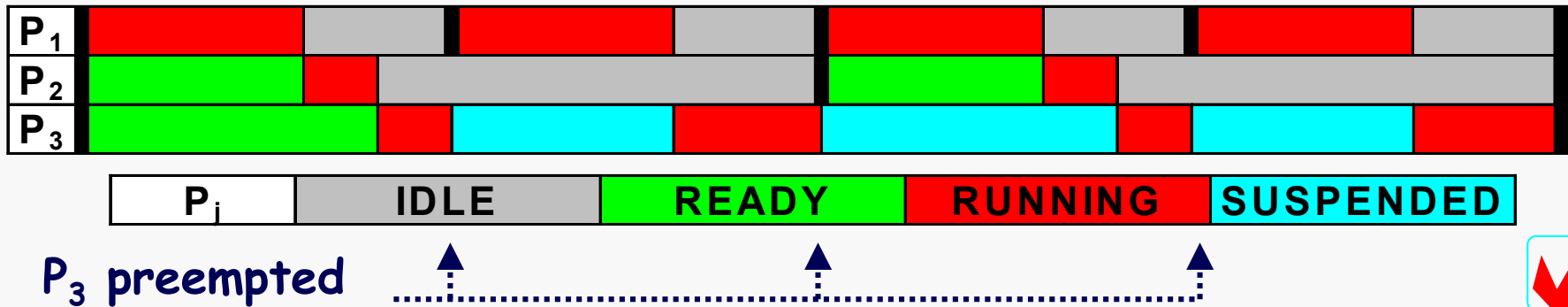
$$U = \sum_{j=1}^N U_j = \sum_{j=1}^N \frac{C_j}{T_j} \leq 1$$

Il  $j^{\text{mo}}$  termine della sommatoria  $C_j / T_j = (C_j (T / T_j)) / T$   
rappresenta la frazione dell'iperperiodo  $T = \text{mcm}(T_1, T_2, \dots, T_N)$   
richiesta per l'esecuzione di  $P_j$ .

# LA STRATEGIA RMPO ...

$A_3$	T[t.u.]	C[t.u.]	C/T	p
$P_1$	25	15	0.6	max
$P_2$	50	5	0.1	
$P_3$	100	30	0.3	min

$$U(A_3) = 1$$



La schedulabilità è garantita se per ogni processo è rispettata la prima deadline a partire dalla condizione più sfavorevole di contemporanea attivazione (istante critico).

$U \leq 1$  è condizione sufficiente per la schedulabilità di un insieme di processi "semplicemente periodici" (con periodi in relazione armonica).

# ... LA STRATEGIA RMPO

Quando non sussiste una relazione armonica tra i periodi dei processi, la condizione  $U \leq 1$  non è sufficiente per garantire la schedulabilità.

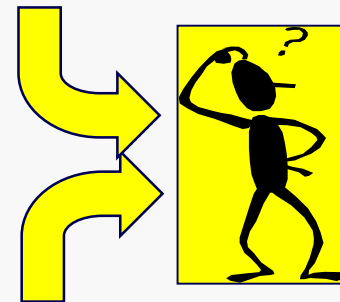
$A_4$	T[t.u.]	C[t.u.]	C/T	p
P <sub>1</sub>	10	5	0.5	max
P <sub>2</sub>	15	6	0.4	min

$$U(A_4) = 0.9$$



$A_5$	T[t.u.]	C[t.u.]	C/T	p
P <sub>1</sub>	10	5	0.50	max
P <sub>2</sub>	19	8	0.42	min

$$U(A_5) = 0.92$$



# OTTIMALITÀ DELLA STRATEGIA RMPO

Se un insieme di processi periodici non è schedabile con RMPO, allora tale insieme non è schedabile con alcun altro algoritmo che preveda un'attribuzione statica di priorità.

## ANALISI DI SCHEDULABILITÀ

- costruzione di diagrammi temporali che evidenziano l'evoluzione dei processi dal punto di vista dell'esecuzione
- applicazione di criteri basati sul fattore di utilizzazione del processore da parte dei (singoli) processi
- calcolo dei tempi di risposta dei processi

Esito sempre conclusivo?



sì



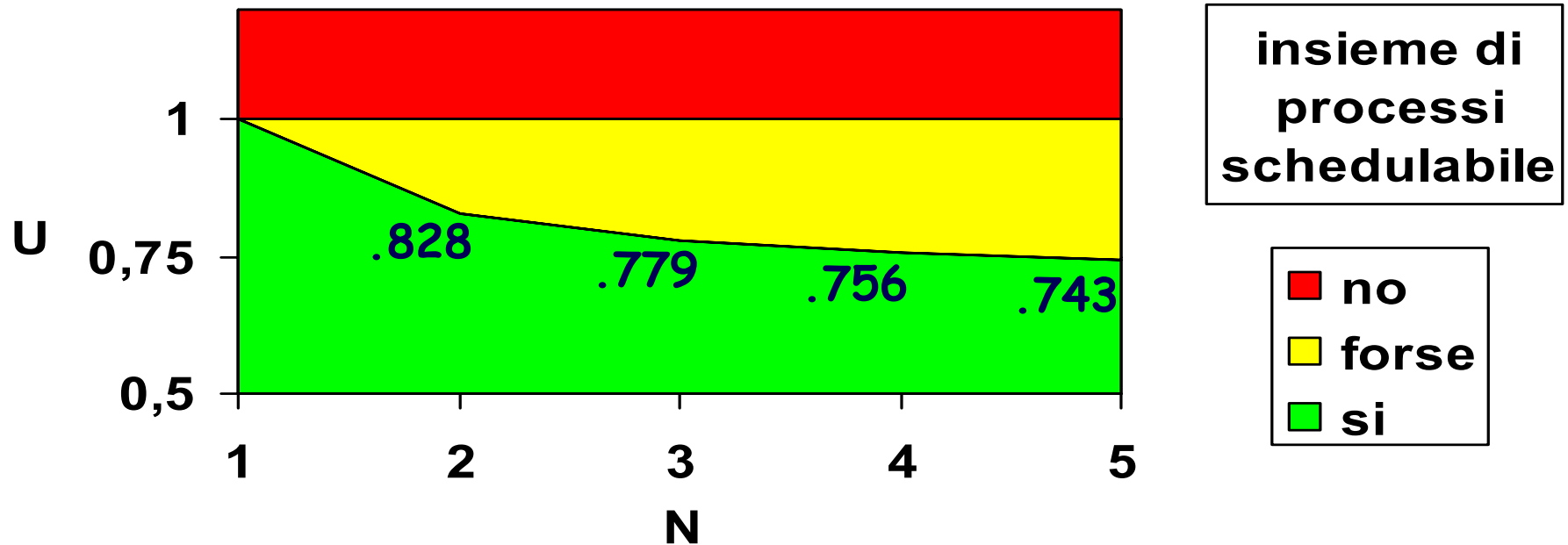
no



# TEST DI SCHEDULABILITÀ BASATI SUL FATTORE DI UTILIZZAZIONE DEL PROCESSORE: TEOREMA DI LIU-LAYLAND (73)

Condizione sufficiente (ma non necessaria) affinché un insieme di  $N$  processi sia schedulabile con la strategia RMPO è che:

$$U \leq U_{\text{RMPO}}(N) = N(2^{1/N} - 1)$$



$$\lim_{N \rightarrow \infty} U_{\text{RMPO}}(N) = \ln 2 = 0.69$$

# UN COROLLARIO DEL TEOREMA

Esplicitando l'“utilizzo del processore schedabile con RMPO”  
in funzione dei singoli fattori di utilizzazione dei processi  
(non semplicemente del numero di processi, come nel teorema),  
si perviene ad un test di schedabilità meno conservativo:

$$U_{\text{RMPO}}(U_1, U_2, \dots, U_N) = \prod_{j=1}^N (1 + U_j) \leq 2$$

	T[t.u.]	C[t.u.]	U
P <sub>1</sub>	10	5	0.5
P <sub>2</sub>	25	5	0.2
P <sub>3</sub>	50	5	0.1

$$\sum_{j=1}^3 U_j = 0.8 > U_{\text{RMPO}}(N=3) = 0.78$$



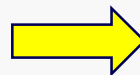
$$U_{\text{RMPO}}(U_1 = 0.5, U_2 = 0.2, U_3 = 0.1) = \prod_{j=1}^3 (1 + U_j) = 1.98 \leq 2$$



# ALTRI TEST DI SCHEDULABILITÀ DIPENDENTI DAI PARAMETRI DEI PROCESSI: IL TEST DI KUO-MOK (91) ...

Un insieme  $S$  di  $N$  processi periodici  $P_1, P_2, \dots, P_N$  è schedulabile con RMPO se (condizione solo sufficiente) l'insieme  $S'$  di  $H < N$  processi periodici  $P_1', P_2', \dots, P_H'$ , ciascuno corrispondente ad un sottoinsieme disgiunto di processi semplicemente periodici in  $S$ , è contraddistinto da un fattore di utilizzazione del processore  $U' \leq U_{RMPO}(H)$ . Al processo  $P_j'$  ( $j = 1, 2, \dots, H$ ) in  $S'$ , equivalente al sottoinsieme di processi  $P_x, P_y, \dots, P_z$  in  $S$ , corrisponde un periodo  $T_j' = \min(T_x, T_y, \dots, T_z)$  e un fattore di utilizzazione del processore  $U_j' = U_x + U_y + \dots + U_z$ .

S	T[t.u.]	C[t.u.]	U
$P_1$	10	5	0.5
$P_2$	25	5	0.2
$P_3$	50	5	0.1



S'	T'[t.u.]	C'[t.u.]	U'
$P_1'$	10	5	0.5
$P_2'$	25	7.5	0.3

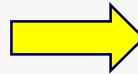
$$\sum_{j=1}^3 U_j = 0.8 > U_{RMPO}(N=3) = 0.78$$

$$\sum_{j=1}^2 U_j' = 0.8 \leq U_{RMPO}(N=2) = 0.828$$

# ... IL TEST DI KUO-MOK

S	T[t.u.]	C[t.u.]	U
P <sub>1</sub>	10	4	0.40
P <sub>2</sub>	20	4	0.20
P <sub>3</sub>	40	8	0.20
P <sub>4</sub>	45	3.6	0.08
P <sub>5</sub>	90	1.8	0.02

Kuo-Mok



S'	T'[t.u.]	C'[t.u.]	U'
P <sub>1</sub> '	10	8	0.8
P <sub>2</sub> '	45	4.5	0.1



$$\sum_{j=1}^5 U_j = 0.9 > U_{\text{RMPO}} (N=5) = 0.743$$



$$\sum_{j=1}^2 U_j' = 0.9 > U_{\text{RMPO}} (N=2) = 0.828$$

corollario

$$U_{\text{RMPO}} (U_1 = 0.4, \dots, U_5 = 0.02) =$$

$$= \prod_{j=1}^5 (1 + U_j) = 2.221 > 2$$



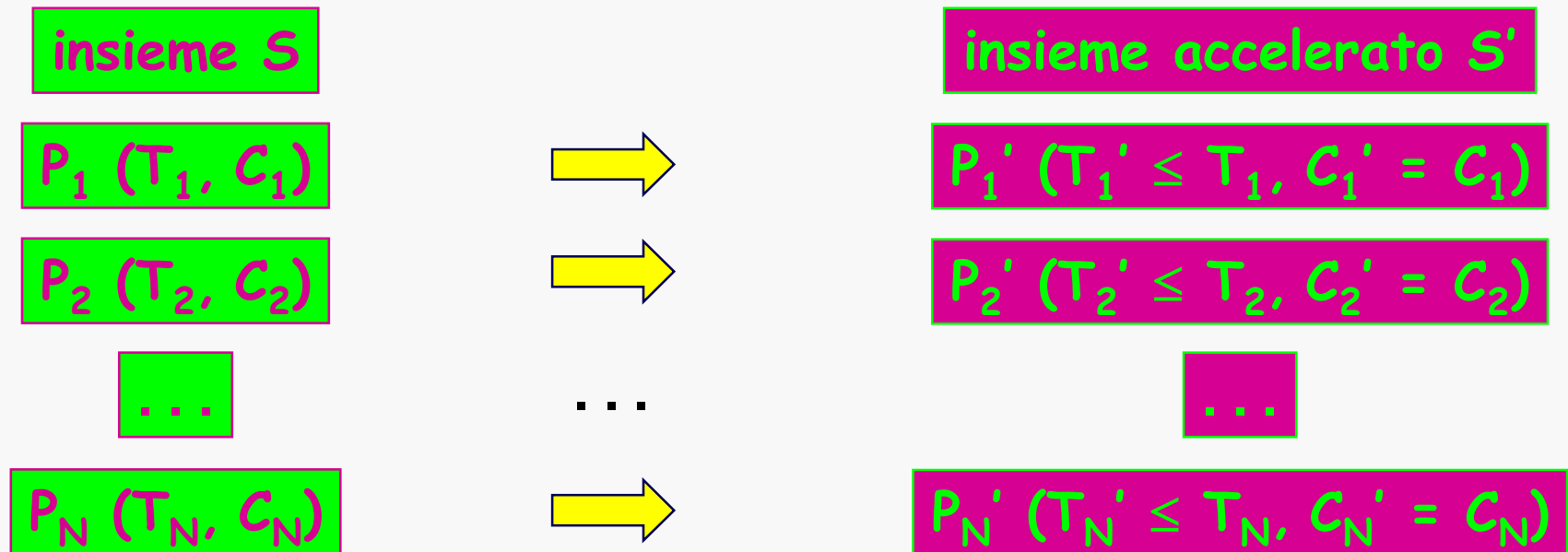
$$U_{\text{RMPO}} (U_1' = 0.8, U_2' = 0.1) =$$

$$= \prod_{j=1}^2 (1 + U_j') = 1.98 \leq 2$$



# ALTRI TEST DI SCHEDULABILITÀ DIPENDENTI DAI PARAMETRI DEI PROCESSI: IL TEST DI HAN (97) ...

Un insieme  $S$  di  $N$  processi periodici  $P_1, P_2, \dots, P_N$  è schedulabile con RMPO se (condizione solo sufficiente) ad esso corrisponde un "insieme accelerato"  $S'$  di  $N$  processi  $P_1', P_2', \dots, P_N'$  semplicemente periodici e con fattore di utilizzazione del processore  $U = U_1' + U_2' + \dots + U_N' \leq 1$ .



# ... IL TEST DI HAN

S	T[t.u.]	C[t.u.]	C/T
P <sub>1</sub>	10	5	0.500
P <sub>2</sub>	16	6	0.375

$$\sum_{j=1}^2 U_j = 0.875 > U_{\text{RMPO}} (N=2) = 0.828$$



$$\prod_{j=1}^2 (1 + U_j) = 2.0625 \geq U_{\text{RMPO}} (U_1 = 0.5, U_2 = 0.375) = 2$$



insieme  
accelerato

S'	T'[t.u.]	C'[t.u.]	U'
P <sub>1</sub> '	8	5	0.625
P <sub>2</sub> '	16	6	0.375

$$U_1' + U_2' \leq 1$$



# ANALISI DI SCHEDULABILITÀ BASATA SUL CALCOLO DEI TEMPI DI RISPOSTA [AUDSLEY et al. (93)]

La schedulabilità è garantita se il tempo di risposta di ogni processo, nelle condizioni più sfavorevoli (cioè a partire da un istante critico), risulta compatibile con la relativa deadline.

$$R_i = C_i + I_i(R_i) \leq D_i = T_i \quad i = 1, 2, \dots, N$$

$I_i$  rappresenta l'interferenza sul tempo di risposta  $R_i$  del processo  $P_i$  derivante dall'esecuzione di processi di priorità superiore  $ps(i)$ .

$$I_i(R_i) = \sum_{j \in ps(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$

Il tempo di risposta di ogni processo può essere calcolato ricorsivamente nel seguente modo:

$$R_i^0 = C_i, \quad R_i^n = C_i + I_i(R_i^{n-1}) \quad n = 1, 2, \dots$$

# ANALISI DI SCHEDULABILITÀ CON L'ALGORITMO DI AUDSLEY

$A_5$	T[t.u.]	C[t.u.]	C/T	p
$P_1$	10	5	0.50	max
$P_2$	19	8	0.42	min

processo  $P_1$  :  $R_1^1 = R_1^0 = C_1 = 5 < D_1 = 10 \Rightarrow$  processo schedulabile  
(il processo a priorità massima è sempre schedulabile

dal momento che non subisce interferenza da parte degli altri processi)

processo  $P_2$  :  $R_2^0 = C_2 = 8$

$$R_2^1 = C_2 + \lceil R_2^0 / T_1 \rceil * C_1 = 8 + \lceil 8/10 \rceil * 5 = 13$$

$$R_2^2 = C_2 + \lceil R_2^1 / T_1 \rceil * C_1 = 8 + \lceil 13/10 \rceil * 5 = 18$$

$$R_2^3 = C_2 + \lceil R_2^2 / T_1 \rceil * C_1 = 8 + \lceil 18/10 \rceil * 5 = 18$$

$R_2^3 = R_2^2 = 18 < D_2 = 19 \Rightarrow$  processo schedulabile

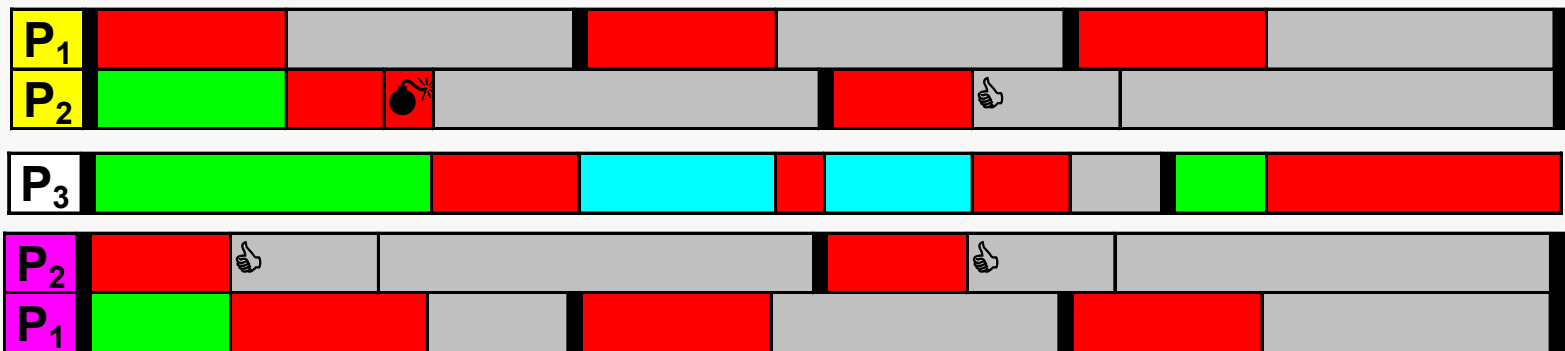
n=0	$C_2 (8) = R_2^0$	
n=1	$I_2 \lceil R_2^0 \rceil (5)$	$+ C_2 (8) = R_2^1 (13)$
n=2	$I_2 \lceil R_2^1 \rceil (10)$	$+ C_2 (8) = R_2^2 (18)$
n=3	$I_2 \lceil R_2^2 \rceil (10)$	$+ C_2 (8) = R_2^3 (18)$



# LA STRATEGIA DMPO

$A_6$	T[t.u.]	C[t.u.]	D[t.u.]	C/T	p(RM)	p(DM)
$P_1$	10	4	10	0.40	max	
$P_2$	15	3	6	0.20		max
$P_3$	22	6	22	0.27	min	min

$$U(A_6) = 0.87$$



## OTTIMALITÀ DELLA STRATEGIA

Se un insieme di processi periodici e/o sporadici non è schedulabile con DMPO, allora tale insieme non è schedulabile con alcun altro algoritmo che preveda un'attribuzione statica di priorità.

# ANALISI DI SCHEDULABILITÀ

Test basati sul fattore di utilizzazione del processore:

Estensione del test di LIU-LAYLAND



$$\sum_{j=1}^N \frac{C_j}{D_j} \leq U_{\text{RMPO}}(N) = N(2^{1/N} - 1)$$

Il test di LEHOCZKY



$$\sum_{j=1}^N \frac{C_j}{T_j} \leq U(N, \delta) = \begin{cases} N((2\delta)^{1/N} - 1) + 1 - \delta & 0.5 \leq \delta \leq 1 \\ \delta & 0 \leq \delta \leq 0.5 \end{cases} \quad \delta = \min_j \{D_j / T_j\}$$

Il test basato sul "fattore di utilizzazione efficace" dei processi

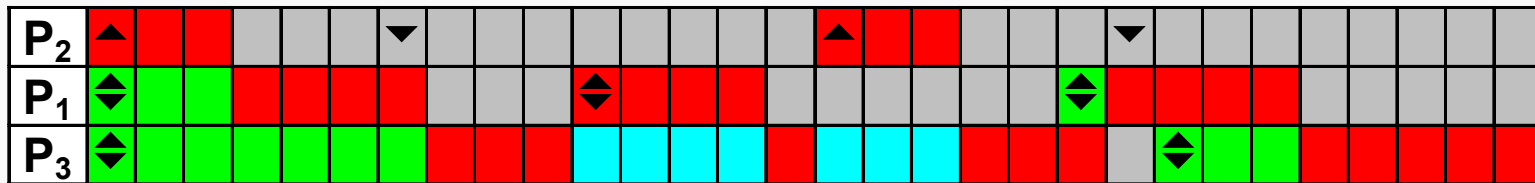
Analisi dei tempi di risposta dei processi:

$$R_j = C_j + I_j(R_j) \leq D_j \quad j = 1, 2, \dots, N$$

# LA STRATEGIA EDF

$A_7$	T[t.u.]	C[t.u.]	D[t.u.]	C/T
$P_1$	10	4	10	0.40
$P_2$	15	3	6	0.20
$P_3$	22	7	22	0.32

$$U(A_7) = 0.92$$



## ANALISI DI SCHEDULABILITÀ

Condizione necessaria e sufficiente affinché un insieme di N processi periodici e/o sporadici sia schedulabile con l'algoritmo EDF è che:

$$U = \sum_{j=1}^N U_j = \sum_{j=1}^N \frac{C_j}{T_j} \leq 1$$



# STRATEGIE PER LA SCHEDULAZIONE DI PROCESSI APERIODICI

- servizio in background
- servizio tramite server

- ◆ a priorità statica

- polling server
- deferrable server
- priority exchange server
- sporadic server
- ...

- ◆ a priorità dinamica

- constant utilization server
- total bandwidth server
- ...

- altri algoritmi

- ◆ a priorità statica

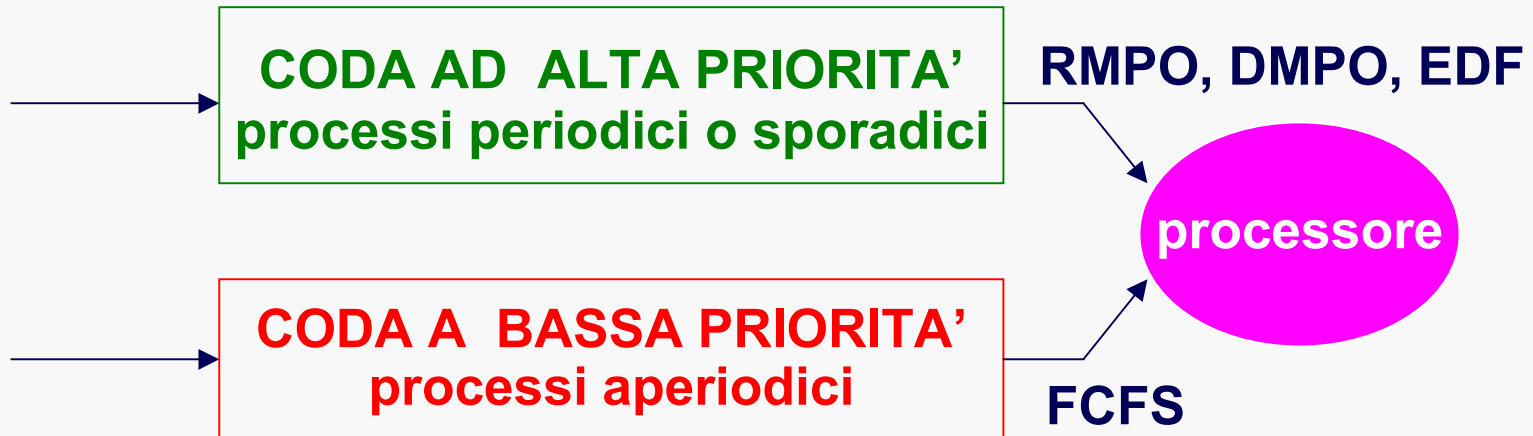
- slack stealer
- ...

- ◆ a priorità dinamica

- ...

# SERVIZIO IN BACKGROUND ...

Le richieste aperiodiche, caratterizzate da vincoli temporali di tipo soft o non real-time, vengono servite solo se non vi sono processi hard real-time (periodici e/o sporadici) da eseguire.



La strategia di schedulazione dei processi aperiodici (tipicamente FCFS) è indipendente da quella selezionata per gli altri processi (tipicamente RMPO, DMPO o EDF).

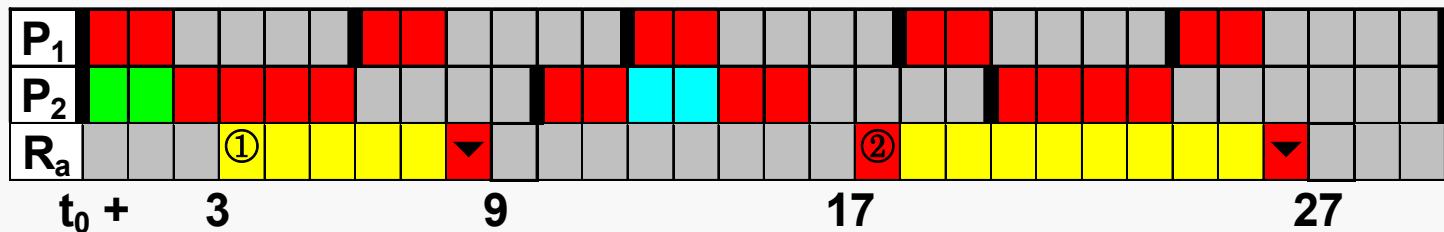
La schedulabilità dei processi hard real-time non è influenzata dal servizio dei processi aperiodici.

# ... SERVIZIO IN BACKGROUND

$N = 2$  processi periodici

$P_1 : T_1 = 6, C_1 = 2$     $P_2 : T_2 = 10, C_2 = 4$

strategia di schedulazione RMPO



$P_j$    IDLE   READY   RUNNING   SUSPENDED

$R_a$    NONE   PENDING   SERVED

I tempi di risposta delle richieste aperiodiche possono risultare notevoli, soprattutto se il fattore di utilizzazione del processore da parte dei processi hard real-time è elevato.

# SERVIZIO TRAMITE SERVER A PRIORITÀ STATICA



Il server è un processo periodico ( $P_S$ ) con periodo  $T_S$  e tempo massimo di esecuzione  $C_S$  (capacità massima del server) prefissati.

$P_S$  è schedulato secondo la stessa strategia (tipicamente, e nel seguito, RMPO) selezionata per la gestione dei processi periodici, in base alla priorità  $p(P_S) \propto 1 / T_S$ , di norma elevata, che gli compete. I tempi di risposta dei processi aperiodici, mediamente sensibilmente inferiori rispetto a quelli derivanti dal servizio in background, dipendono dalle regole, peculiari di ciascuna tipologia di server, secondo cui è operato il riempimento ed il consumo della capacità del server stesso.



# SERVIZIO TRAMITE DEFERRABLE SERVER [Lehoczky, Sha, Strosnider (87)]

Regole di riempimento e di consumo della capacità di  $P_s$ .  
 La capacità viene ripristinata al valore  $C_s$  all'inizio di ogni periodo  $T_s$ .  
 La capacità disponibile è conservata in assenza di richieste aperiodiche pendenti: il servizio può pertanto essere differito (bandwidth-preserving server). La capacità viene progressivamente consumata durante il servizio di richieste aperiodiche.

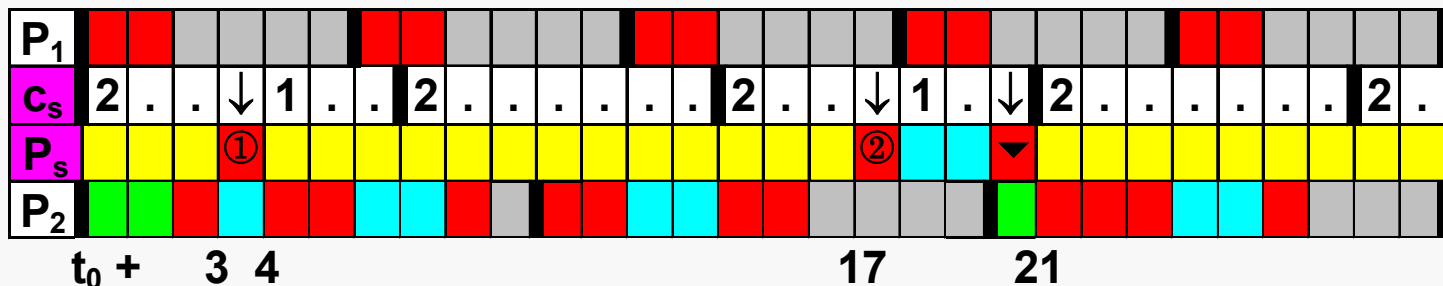
$N = 2$  processi periodici

$P_1 : T_1 = 6, C_1 = 2$

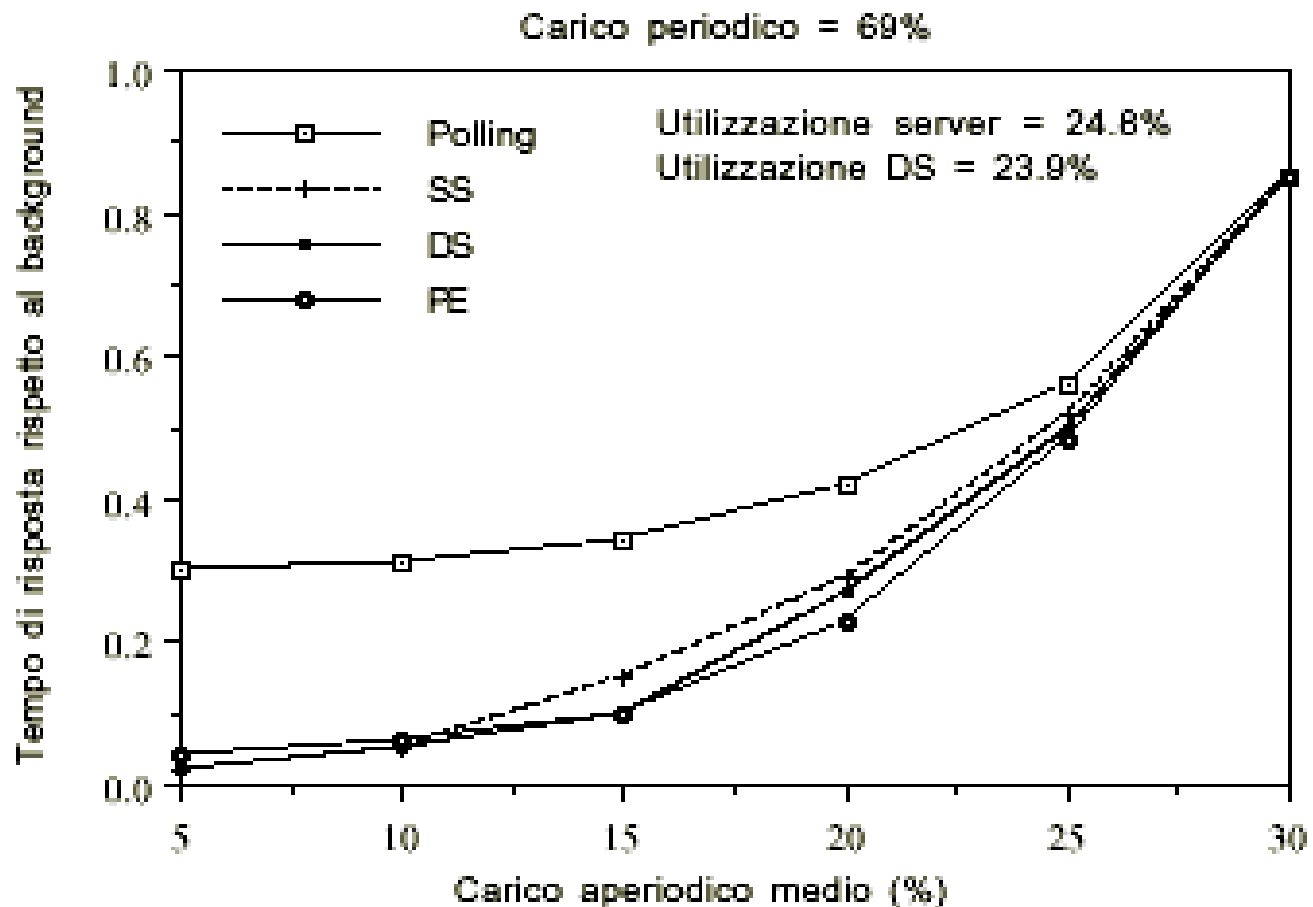
$P_2 : T_2 = 10, C_2 = 4$

$P_s : T_s = 7, C_s = 2$

strategia di schedulazione RMPO



# Confronto delle prestazioni: risultati di una simulazione disponibili in rete



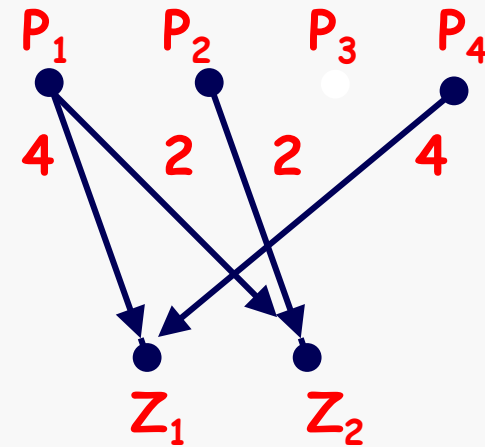
# PROTOCOLLI DI ACCESSO A RISORSE CONDIVISE

## Insiemi di regole che definiscono

- quando ed in quali condizioni ciascuna richiesta di accesso viene accettata
- come un insieme di processi con risorse condivise vengono schedati

$A_8$	p		C [t.u.] - C <sub>type</sub>					
P <sub>1</sub>	max	6	■	■	■	■	■	■
P <sub>2</sub>	.	4	■	■	■	■		
P <sub>3</sub>	.	4	■	■	■	■		
P <sub>4</sub>	min	6	■	■	■	■	■	■

"Resource Requirements Graph (RRG)"



## applicazione A<sub>8</sub>

- ▼ 4 processi periodici P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, P<sub>4</sub> con priorità p<sub>1</sub> > p<sub>2</sub> > p<sub>3</sub> > p<sub>4</sub>
- ▼ 2 risorse condivise Z<sub>1</sub> (da parte di P<sub>1</sub> e P<sub>4</sub>) e Z<sub>2</sub> (da parte di P<sub>1</sub> e P<sub>2</sub>)
  - P<sub>1</sub>: C = 6 t.u., accesso in sequenza a Z<sub>1</sub> (4) e Z<sub>2</sub> (2)
  - P<sub>2</sub>: C = 4 t.u., esecuzione normale (2) con accesso intermedio a Z<sub>2</sub> (2)
  - P<sub>3</sub>: C = 4 t.u., esecuzione normale senza accesso a Z<sub>1</sub> e Z<sub>2</sub>
  - P<sub>4</sub>: C = 6 t.u., esecuzione normale (2) con accesso intermedio a Z<sub>1</sub> (4)



# IL PROTOCOLLO PRIORITY INHERITANCE [CORNHILL et al. (87)]

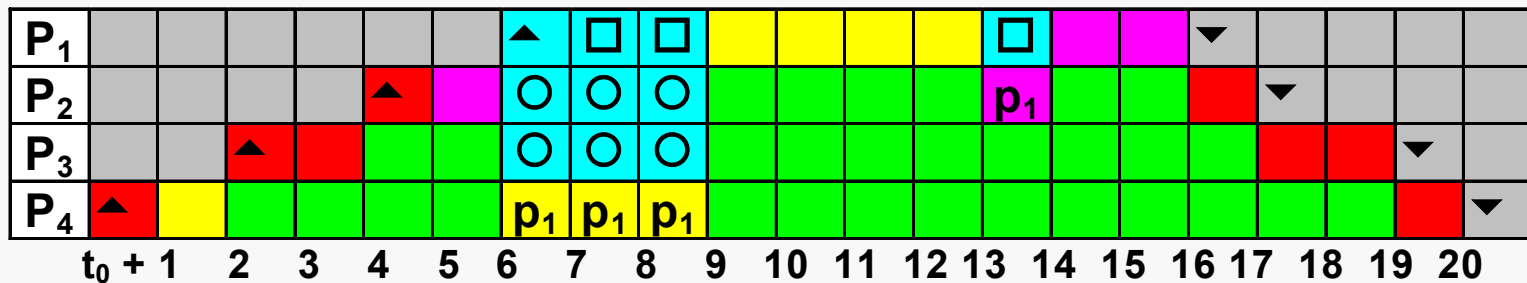
Un processo  $P_L$  eredita la "priorità corrente" di un processo  $P_H$  da esso direttamente bloccato. Al rilascio della risorsa, il processo  $P_L$  riassume la priorità che aveva all'atto dell'acquisizione della risorsa.

$A_8$	$p$	$a[t.u.]$	$C[t.u.] - C_{type}$						$R[t.u.]$
$P_1$	max	6	6	□	□	□	□	□	10
$P_2$	.	4	4	□	□	□	□		13
$P_3$	.	2	4	□	□	□	□		17
$P_4$	min	0	6	□	□	□	□	□	20

blocking

□ direct

○ push-through



$P_i$	IDLE	READY	BLOCKED
	RUNNING	LOCKING $Z_1$	LOCKING $Z_2$

Non implica la conoscenza a priori delle risorse utilizzate dai singoli processi. Previene inversioni di priorità incontrollate, ma non concatenazioni di blocchi o deadlock. OdI 16/11/04 39

# IL PROTOCOLLO IMMEDIATE CEILING PRIORITY (ICP) [SHA et al. (90)] ...

Ad ogni processo è associata staticamente una priorità di default.

Ad ogni risorsa è associata staticamente la massima priorità dei processi che ad essa possono accedere (tetto di priorità).

Un processo può accedere ad una risorsa soltanto se la sua priorità è più alta del tetto di priorità di tutte le risorse al momento utilizzate da altri processi.

Ad ogni processo è associata dinamicamente la priorità più alta fra la propria priorità statica ed il massimo tetto di priorità associato a tutte le risorse in suo possesso.

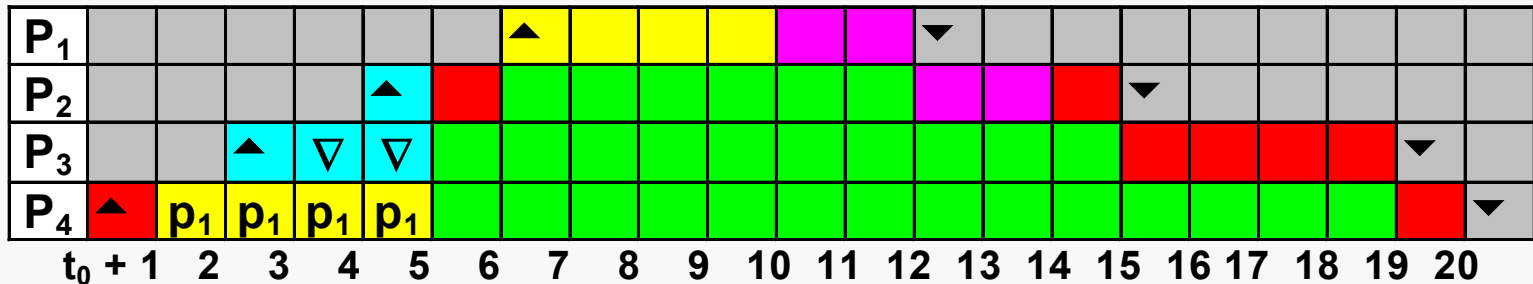
Tutte le risorse utilizzate da un processo sono libere allorché ha effettivamente inizio la sua esecuzione.

Implica la conoscenza a priori delle risorse utilizzate dai singoli processi. Previene inversioni di priorità incontrollate e deadlock.

# ... IL PROTOCOLLO ICP

$A_8$	$p$	$a[t.u.]$	$C[t.u.] - C_{type}$	$R[t.u.]$
$P_1$	max	6	6	6
$P_2$	.	4	4	11
$P_3$	.	2	4	17
$P_4$	min	0	6	20

tetto di priorità ( $Z_1$ ) =  $p_1$   
 tetto di priorità ( $Z_2$ ) =  $p_1$



- blocking
- direct
- push-through
- ▽ ceiling