

# PROVA SCRITTA DI SISTEMI IN TEMPO REALE (15/9/2017)

## PROBLEMA n. 1

L'esecuzione del seguente insieme di processi periodici, tra loro indipendenti, è affidata ad un sistema di elaborazione monoprocessore:

	$T_i$ [t.u.]	$C_i$ [t.u.]
$P_1$	10	3
$P_2$	15	3
$P_3$	20	6
$P_4$	60	6

A) Schedulazione "clock-driven"

1. Si identifichi un feasible schedule per l'insieme di processi in accordo all'approccio Cyclic Executive, selezionando per il ciclo minore, in caso di più alternative ammissibili, la dimensione massima:

- (a) dimensionamento del ciclo maggiore

$$M = \text{mcm}(T_1, T_2, T_3, T_4) = 60$$

- (b) dimensionamento del ciclo minore

$$\max(C_1, C_2, C_3, C_4) = 6 \leq m \leq 10 = \min(T_1, T_2, T_3, T_4) \rightarrow m = 6 \dots 10$$

$$M \bmod m = 0 \rightarrow m = 6, 10$$

$$m = 10: 2m - \text{mcd}(m, \{T_1, T_2, T_3, T_4\}) = 20 - \{10, 5, 10, 10\} = \{10, 15, 10, 10\} \leq \{T_1, T_2, T_3, T_4\} \rightarrow \text{OK}$$

$$m = 6: 2m - \text{mcd}(m, \{T_1, T_2, T_3, T_4\}) = 12 - \{2, 3, 2, 6\} = \{10, 9, 10, 6\} \leq \{T_1, T_2, T_3, T_4\} \rightarrow \text{OK}$$

$$m = 10$$

- (c) calcolo del numero dei cicli minori e dei job per processo nell'ambito di ciascun ciclo maggiore

$$n_{cm} = 6, n_{J1} = 6, n_{J2} = 4, n_{J3} = 3, n_{J4} = 1$$

- (d) identificazione del ciclo minore (o dei cicli minori) in cui ciascun job può essere eseguito

	$J_{11}$	$J_{12}$	$J_{13}$	$J_{14}$	$J_{15}$	$J_{16}$	$J_{21}$	$J_{22}$	$J_{23}$	$J_{24}$	$J_{31}$	$J_{32}$	$J_{33}$	$J_{41}$
$c_1$	x						x				x			x
$c_2$		x									x			x
$c_3$			x					x				x		x
$c_4$				x					x			x		x
$c_5$					x								x	x
$c_6$						x				x			x	x

- (e) pianificazione dell'esecuzione di ciascun job

c <sub>1</sub>	J <sub>11</sub>	J <sub>21</sub>				
c <sub>2</sub>	J <sub>12</sub>	J <sub>31</sub>				
c <sub>3</sub>	J <sub>13</sub>	J <sub>22</sub>				
c <sub>4</sub>	J <sub>14</sub>	J <sub>23</sub>				
c <sub>5</sub>	J <sub>15</sub>	J <sub>33</sub>				
c <sub>6</sub>	J <sub>16</sub>	J <sub>24</sub>				

esito positivo per tutti i job

sì	no	quali job ?
	×	J <sub>32</sub> , J <sub>41</sub>

- (f) in caso di esito negativo, rilassamento del minor numero possibile di vincoli

$$P_3 (6, 20) \Rightarrow P_3' (3, 20), P_3'' (3, 20) \text{ con } P_3' < P_3''$$

$$P_4 (6, 60) \Rightarrow P_4' (3, 60), P_4'' (3, 60) \text{ con } P_4' < P_4''$$

e iterazione del procedimento

c <sub>1</sub>	J <sub>11</sub>	J <sub>21</sub>	J <sub>31</sub> '	
c <sub>2</sub>	J <sub>12</sub>	J <sub>41</sub> '	J <sub>31</sub> ''	
c <sub>3</sub>	J <sub>13</sub>	J <sub>22</sub>	J <sub>32</sub> '	
c <sub>4</sub>	J <sub>14</sub>	J <sub>23</sub>	J <sub>32</sub> ''	
c <sub>5</sub>	J <sub>15</sub>	J <sub>41</sub> ''	J <sub>33</sub> '	
c <sub>6</sub>	J <sub>16</sub>	J <sub>24</sub>	J <sub>33</sub> ''	

## B) Schedulazione "priority-driven"

1. Si verifichi se l'insieme di processi è schedulabile con l'algoritmo RMPO, ricorrendo all'applicazione, nell'ordine, di uno o più dei seguenti metodi fino ad identificare una risposta conclusiva al riguardo:

- il test di Kuo-Mok ed il corollario del teorema di Liu-Layland

S'	T <sub>i</sub> ' [t.u.]	C <sub>i</sub> ' [t.u.]	U <sub>i</sub> '
P <sub>1</sub> ' ≡ {P <sub>1</sub> , P <sub>3</sub> , P <sub>4</sub> }	10	7	0.7
P <sub>2</sub> ' ≡ {P <sub>2</sub> }	15	3	0.2

$$U = U_1' + U_2' = 0.9 > U_{RMPO} (N=2) = 0.828$$

$$(1 + U_1') * (1 + U_2') = 2.04 > 2$$

esito positivo		risposta conclusiva	
sì	no	sì	no
	×		×

- il test di Han

S'	T <sub>i</sub> ' [t.u.]	C <sub>i</sub> [t.u.]	U <sub>i</sub> '
P <sub>1</sub> ' ≡ P <sub>1</sub>	10	3	0.3
P <sub>2</sub> '	10	3	0.3
P <sub>3</sub> ' ≡ P <sub>3</sub>	20	6	0.3
P <sub>4</sub> ' ≡ P <sub>4</sub>	60	6	0.1

$$U = U_1' + U_2' + U_3' + U_4' = 1 \leq 1$$

esito positivo		risposta conclusiva	
sì	no	sì	no
×		×	

- l'algoritmo di Audsley

	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>
R <sub>i</sub> <sup>0</sup> = C <sub>i</sub>	3	3	6	6
R <sub>i</sub> <sup>1</sup> = C <sub>i</sub> + I <sub>i</sub> (R <sub>i</sub> <sup>0</sup> )	3 + 0 = 3	3 + 3 = 6	6 + 3 + 3 = 12	6 + 3 + 3 + 6 = 18
R <sub>i</sub> <sup>2</sup> = C <sub>i</sub> + I <sub>i</sub> (R <sub>i</sub> <sup>1</sup> )		3 + 3 = 6	6 + 6 + 3 = 15	6 + 6 + 6 + 6 = 24
R <sub>i</sub> <sup>3</sup> = C <sub>i</sub> + I <sub>i</sub> (R <sub>i</sub> <sup>2</sup> )			6 + 6 + 3 = 15	6 + 9 + 6 + 12 = 33
R <sub>i</sub> <sup>4</sup> = C <sub>i</sub> + I <sub>i</sub> (R <sub>i</sub> <sup>3</sup> )				6 + 12 + 9 + 12 = 39
R <sub>i</sub> <sup>5</sup> = C <sub>i</sub> + I <sub>i</sub> (R <sub>i</sub> <sup>4</sup> )				6 + 12 + 9 + 12 = 39

esito positivo		risposta conclusiva	
sì	no	sì	no
×		×	

2. Nell'ipotesi che P<sub>3</sub> e P<sub>4</sub> abbiano rispettivamente deadline ([t.u.]) D<sub>3</sub> = 16 e D<sub>4</sub> = 32, si verifichi se l'insieme di processi è schedabile con l'algoritmo DMPO, ricorrendo all'applicazione, nell'ordine, di uno o più dei seguenti metodi fino ad identificare una risposta conclusiva al riguardo:

- il test basato sul fattore di utilizzazione efficace del processore

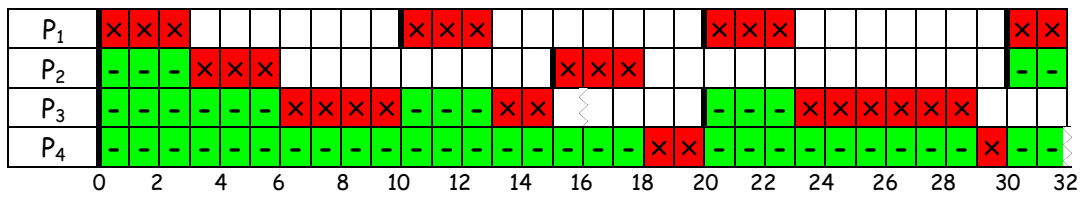
	H <sub>n</sub>	H <sub>1</sub>	f	U(n, δ)	f ≤ U(n, δ)
P <sub>1</sub>	{ }	{ }	3/10=0.3	U(1,1)=1	sì
P <sub>2</sub>	{P <sub>1</sub> }	{ }	3/10+3/15=0.5	U(2,1)=0.828	sì
P <sub>3</sub>	{P <sub>1</sub> , P <sub>2</sub> }	{ }	3/10+3/15+6/20=0.8	U(3,0.8)=0.709	no
P <sub>4</sub>	{P <sub>1</sub> , P <sub>2</sub> , P <sub>3</sub> }	{ }	3/10+3/15+6/20+6/60=0.9	U(4,0.533)=0.531	no

esito positivo		risposta conclusiva	
sì	no	sì	no
	×		×

- il diagramma temporale che evidenzia con riferimento alla seguente notazione

	P <sub>i</sub>
	Idle
-	Ready
×	Running

l'evoluzione dei processi dal punto di vista dell'esecuzione nelle condizioni più sfavorevoli



esito positivo		risposta conclusiva	
sì	no	sì	no
	×	×	

3. Nelle stesse ipotesi di cui al punto precedente, si verifichi se l'insieme di processi è schedabile con l'algoritmo EDF ricorrendo all'applicazione, nell'ordine, di uno o più dei seguenti metodi fino ad identificare una risposta conclusiva al riguardo:

- il test basato sulla densità di utilizzazione del processore

	$T_i$ [t.u.]	$C_i$ [t.u.]	$D_i$ [t.u.]	$C_i/D_i$
$P_1$	10	3	10	0.3
$P_2$	15	3	15	0.2
$P_3$	20	6	16	0.375
$P_4$	60	6	32	0.1875

$$\Delta = C_1/D_1 + C_2/D_2 + C_3/D_3 + C_4/D_4 = 1.0625 > 1$$

esito positivo		risposta conclusiva	
sì	no	sì	no
	×	×	

- l'approccio "processor demand"

$$t^* = ((1-16/20)*6 + (1-32/60)*6) / (1-0.9) = 40$$

n	$BI^n$
0	$3 + 3 + 6 + 6 = 18$
1	$6 + 6 + 6 + 6 = 24$
2	$9 + 6 + 12 + 6 = 33$
3	$12 + 9 + 12 + 6 = 39$
4	$12 + 9 + 12 + 6 = 39$

$$BI = 39$$

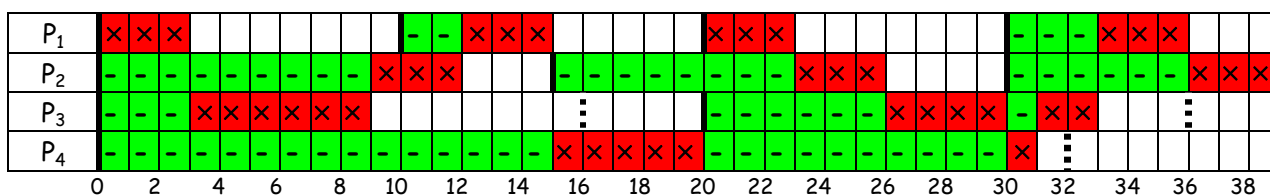
$$\min\{t^*, BI\} = 39 \quad \mathcal{D}^* \cap \mathcal{D} = \{10, 15, 16, 20, 30, 32, 36\}$$

t	$C_1(0,t)$	$C_2(0,t)$	$C_3(0,t)$	$C_4(0,t)$	$C_p(0,t)$	$\leq t$
10	3	0	0	0	3	sì
15	3	3	0	0	6	sì
16	3	3	6	0	12	sì
20	6	3	6	0	15	sì

30	9	6	6	0	21	sì
32	9	6	6	6	27	sì
36	9	6	12	6	33	sì

4. Nelle stesse ipotesi di cui al punto precedente, si verifichi tramite la costruzione di un diagramma temporale se l'insieme di processi è schedulabile con l' algoritmo non strict LST:

slack	t=0	t=10	t=15	t=20	t=30
P <sub>1</sub>	7	7	-	7	7
P <sub>2</sub>	12	3	12	7	12
P <sub>3</sub>	10	-	-	10	4
P <sub>4</sub>	26	16	11	11	1



esito positivo

sì	no
----	----

risposta conclusiva

sì	no
----	----

## PROBLEMA n. 2

Un sistema di elaborazione monoprocessoore ha il compito di eseguire tre processi periodici  $P_1, P_2, P_3$  tra loro indipendenti, gestendo nel contempo tre richieste aperiodiche  $R_{a1}, R_{a2}, R_{a3}$ :

	$T_i$ [t.u.]	$C_i$ [t.u.]
$P_1$	4	1
$P_2$	10	2
$P_3$	32	8

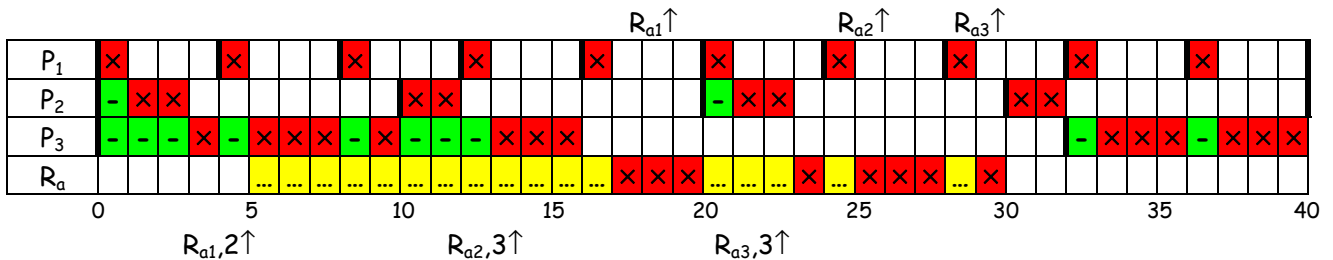
	$a_i$ [t.u.]	$C_i$ [t.u.]
$R_{a1}$	5	2
$R_{a2}$	14	3
$R_{a3}$	23	3

A) Con l'ausilio di diagrammi temporali e facendo riferimento alla seguente notazione

	$P_i$	$R_a$	$S$
	Idle	None	Idle
-	Ready	-	Ready
...	-	Pending	Waiting
x	Running	Being Served	Running

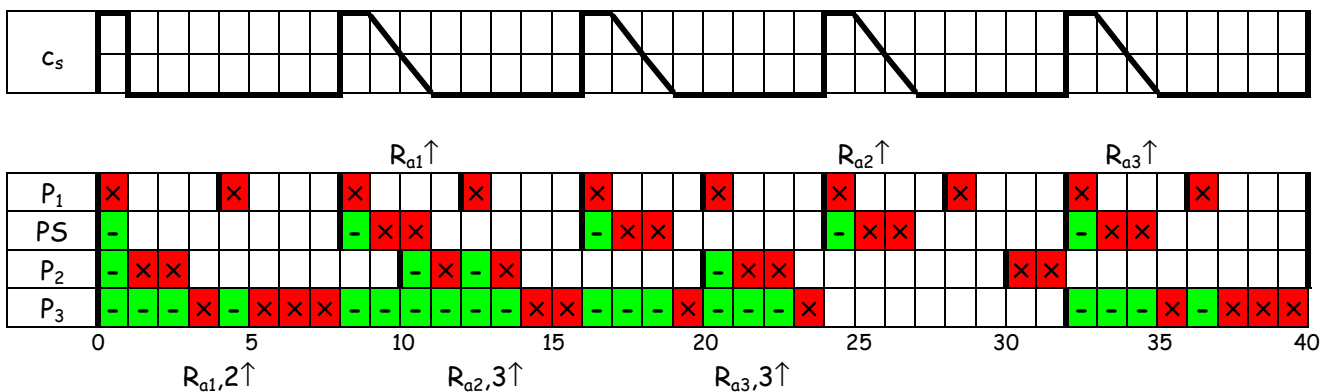
si determini (riportandone i valori nella sottostante tabella) il tempo di completamento della esecuzione  $f_i$  [t.u.] ed il ritardo sul tempo di risposta  $f_i - a_i - C_i$  [t.u.] delle richieste aperiodiche derivanti dalla loro gestione tramite:

1. Servizio in Background (strategia di schedulazione dei processi: RMPO)

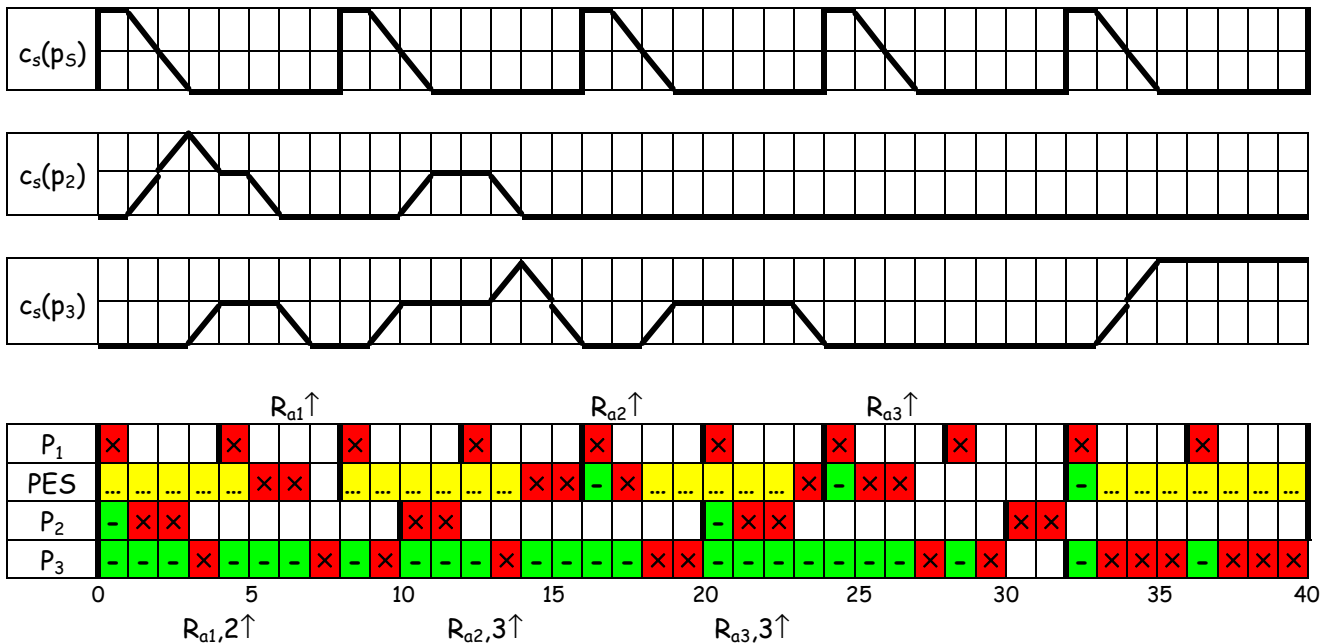


2. Server a priorità statica di periodo  $T_s = 8$  t.u. e capacità  $C_s = 2$  t.u. (strategia di schedulazione dei processi e del Server: RMPO)

• Polling Server



• Priority Exchange Server

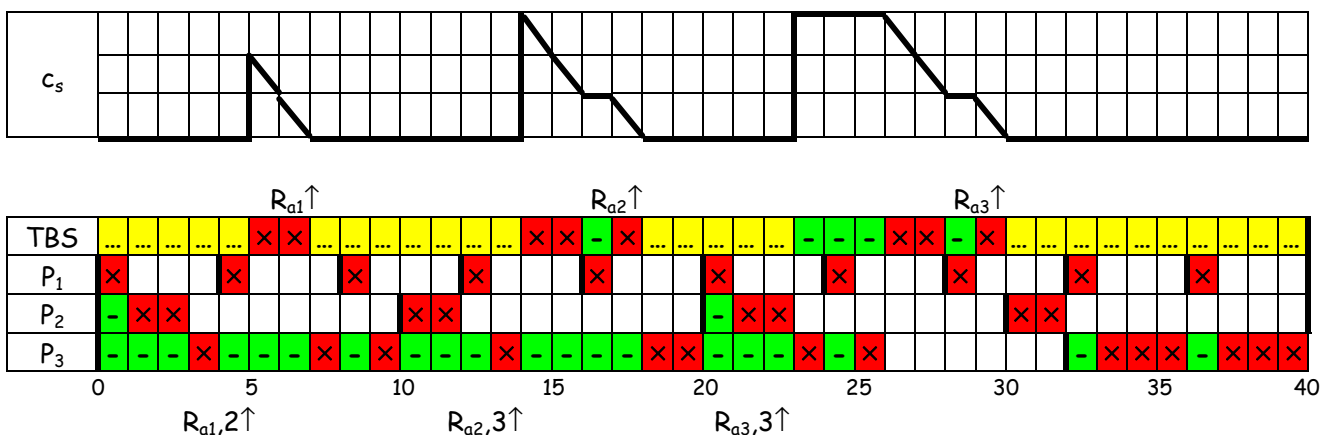


3. Server a priorità dinamica con fattore di utilizzazione del processore  $U_s$  più elevato possibile compatibilmente con la schedulabilità dei processi periodici (strategia di schedulazione dei processi e del Server: EDF)

$$U_s = 1 - U_p = 1 - 0.7 = 0.3$$

• Total Bandwidth Server

	$a_i$ [t.u.]	$C_i$ [t.u.]	$d_{si}$ [t.u.]
$R_{a1}$	5	2	$5 + 2 / 0.3 = 11.7$
$R_{a2}$	14	3	$14 + 3 / 0.3 = 24$
$R_{a3}$	23	3	$24 + 3 / 0.3 = 34$



• Constant Utilization Server

Si verifichi se CUS è in grado di fornire le stesse prestazioni di TBS, motivandone le ragioni ed evidenziando in caso contrario nella tabella sottostante i dati relativi alle richieste aperiodiche il cui tempo di completamento del servizio risulta superiore

$R_{a1}$ :  $f_1(CUS) = f_1(TBS)$   
 $R_{a2}$ :  $a_2 = 14 > d_{s1} = 11.7 \rightarrow f_2(CUS) = f_2(TBS)$   
 $R_{a3}$ :  $a_3 = 23 < d_{s2} = 24$  TBS non è prioritario in [23-24]  $\rightarrow f_3(CUS) = f_3(TBS)$

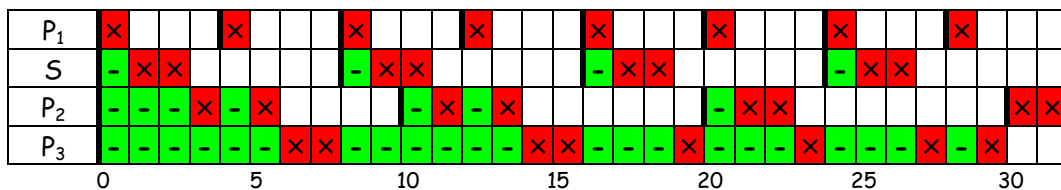
	$a_i$	$C_i$	$f_i$	$f_i - a_i - C_i$	$f_i$	$f_i - a_i - C_i$	$f_i$	$f_i - a_i - C_i$	$f_i$	$f_i - a_i - C_i$	$f_i$	$f_i - a_i - C_i$
$R_{a1}$	5	2	19	12	11	4	7	0	7	0	-	-
$R_{a2}$	14	3	26	7	26	7	18	1	18	1	-	-
$R_{a3}$	23	3	30	4	35	9	27	1	30	4	-	-
			BKG		PS		PES		TBS		CUS	

B) Si verifichi se, indipendentemente dalla effettiva distribuzione temporale e dal tempo di servizio delle richieste aperiodiche, la loro gestione tramite i suddetti server a priorità statica ( $T_s = 8$  t.u.,  $C_s = 2$  t.u.) compromette la schedulabilità  
 (a) dei processi  $P_1, P_2, P_3$ ,  
 (b) di qualunque altro insieme di 3 processi periodici  $P_1', P_2', P_3'$  aventi complessivamente lo stesso fattore di utilizzazione del processore ( $U_1' + U_2' + U_3' = U_1 + U_2 + U_3$ ) e periodi  $T_1', T_2', T_3' > T_s$ , identificando in entrambi i casi il valore massimo che  $C_s$  può assumere, a parità di  $T_s$  (8 t.u.), onde prevenire tale inaccettabile circostanza.

• (a)

sì	no
	×

infatti:



$C_s = 2$  t.u.

• (b)

sì	no
×	

infatti:

$$U_s = 0.3 > 2 / (1 + 0.7 / 3)^3 - 1 = 0.066$$

$$C_s = 8 * 0.066 = 0.53 \text{ t.u.}$$



## PROBLEMA n. 3

L'esecuzione concorrente del seguente insieme di processi

	$T_i$ [t.u.]	$D_i$ [t.u.]	$C_i$ [t.u.]
$P_1$	25	25	4
$P_2$	30	30	3
$P_3$	35	35	15
$P_4$	45	40	3
$P_5$	55	45	4

attivati all'istante  $\phi_i$  ([t.u.]) = 8, 6, 4, 2, 0, rispettivamente, è affidata ad un sistema di elaborazione monoprocesso.

I processi  $P_1, P_2, P_3$  e  $P_5$  condividono 4 risorse  $R_1, R_2, R_3$  e  $R_4$ , con accesso mutuamente esclusivo ( $u_1 = u_2 = u_3 = u_4 = 1$ ) regolamentato dai semafori  $S_1, S_2, S_3$  e  $S_4$ . Più precisamente, l'esecuzione di ciascun job di ogni processo comporta le seguenti elaborazioni, contraddistinte dal tempo indicato tra parentesi:

- $P_1$ : esecuzione normale (1 t.u.), accesso a  $R_2$  (1 t.u.), accesso a  $R_4$  (1 t.u.), esecuzione normale (1 t.u.);
- $P_2$ : esecuzione normale (1 t.u.), accesso a  $R_1$  (1 t.u.), esecuzione normale (1 t.u.);
- $P_3$ : accesso a  $R_1$  (6 t.u. complessivamente) con accesso annidato dopo 1 t.u. a  $R_4$  (4 t.u.), esecuzione normale (1 t.u.), accesso a  $R_2$  (3 t.u. complessivamente) con accesso annidato dopo 1 t.u. a  $R_4$  (1 t.u.), esecuzione normale (1 t.u.), accesso a  $R_3$  (4 t.u. complessivamente) con accesso annidato dopo 1 t.u. a  $R_4$  (2 t.u.);
- $P_4$ : esecuzione normale (3 t.u.);
- $P_5$ : esecuzione normale (1 t.u.), accesso a  $R_3$  (2 t.u.), esecuzione normale (1 t.u.).

	$C_{type}$														
$P_1$		R <sub>2</sub>	R <sub>4</sub>												
$P_2$		R <sub>1</sub>													
$P_3$	R <sub>1</sub>	R <sub>4</sub>	R <sub>4</sub>	R <sub>4</sub>	R <sub>4</sub>	R <sub>1</sub>		R <sub>2</sub>	R <sub>4</sub>	R <sub>2</sub>		R <sub>3</sub>	R <sub>4</sub>	R <sub>4</sub>	R <sub>3</sub>
$P_4$															
$P_5$		R <sub>3</sub>	R <sub>3</sub>												

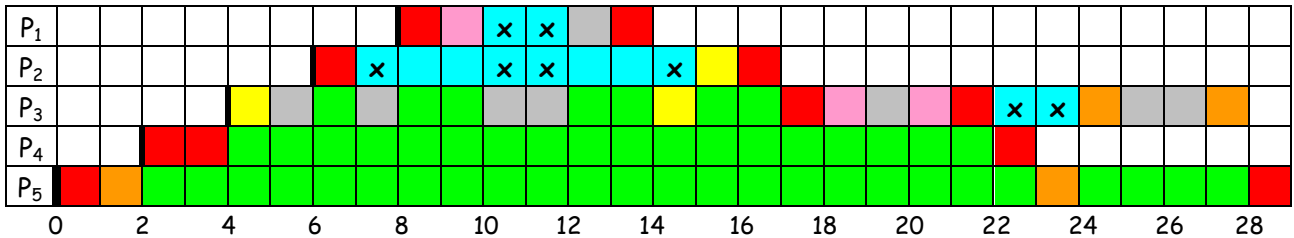
1. Nell'ipotesi che i processi siano schedulati in accordo alla strategia DMPO,

➤ si identifichi con l'ausilio di diagrammi temporali, facendo riferimento alla seguente notazione

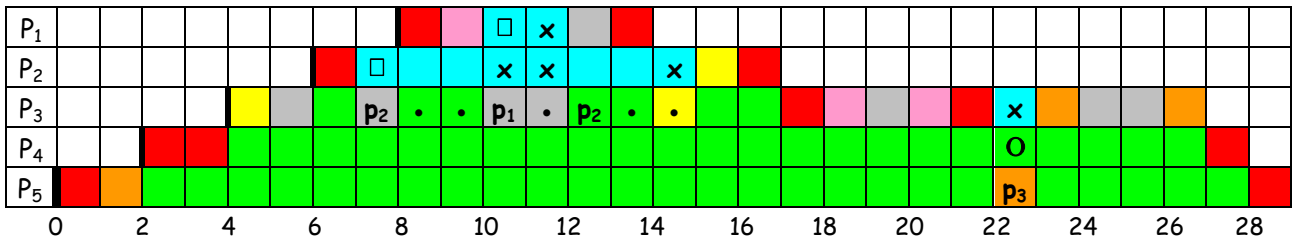
	$P_i$
	Idle
	ready ( $\times [O, \nabla]$ when blocked)
	suspended ( $\times [\square, \nabla]$ when blocked)
	running without locking any semaphore
	executing critical section protected by semaphore $S_1$
	executing critical section protected by semaphore $S_2$
	executing critical section protected by semaphore $S_3$
	executing critical section protected by semaphore $S_4$

ed evidenziando esplicitamente la priorità corrente ( $\pi_i$ ) di un processo soltanto se essa differisce dalla corrispondente priorità nominale ( $p_i$ ), il tempo di completamento dell'esecuzione ( $f_i$  [t.u.]) ed il ritardo sul tempo di risposta ( $f_i - \phi_i - C_i$  [t.u.]) del primo job di ciascun processo (riportandone i valori nella sottostante tabella) nel caso di:

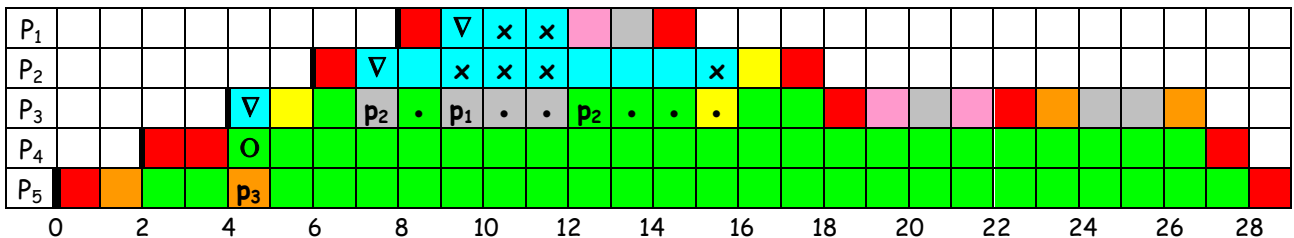
- esecuzione in base alle relative priorità statiche (NOP)



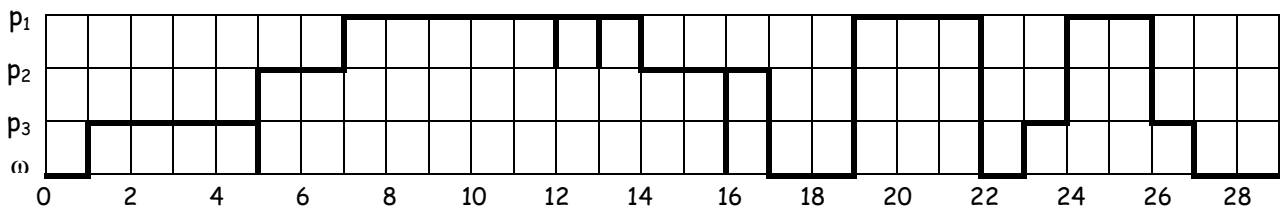
- applicazione del protocollo priority inheritance (PIP)



- applicazione del protocollo priority ceiling (PCP) [ $PC_1 = p_2, PC_2 = p_1, PC_3 = p_3, PC_4 = p_1$ ]



$\Pi \Gamma_s$



	$\phi_i$	$C_i$	$f_i$	$f_i - \phi_i - C_i$	$f_i$	$f_i - \phi_i - C_i$	$f_i$	$f_i - \phi_i - C_i$
$P_1$	8	4	14	2	14	2	15	3
$P_2$	6	3	17	8	17	8	18	9
$P_3$	4	15	28	9	27	8	27	8
$P_4$	2	3	23	18	28	23	28	23
$P_4$	0	4	29	25	29	25	29	25
			NOP		PIP		PCP	

➤ Si determini il massimo tempo di blocco di ogni processo derivante dall'applicazione di PIP, PCP, IPCP

	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>	R <sub>4</sub>	B <sub>i</sub>	
P <sub>1</sub>		1		1	4	4
P <sub>2</sub>	1				6	6
P <sub>3</sub>	6	3	4	4	2	2
P <sub>4</sub>					2	2
P <sub>5</sub>			2		0	0
					PIP	PCP

e si verifichi, applicando l'algoritmo di Audsley, se l'insieme dei processi è schedulabile qualunque sia il protocollo usato.

	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>
R <sub>i</sub> <sup>0</sup>	4 + 4 = 8	3 + 6 = 9	15 + 2 = 17	3 + 2 = 5	4 + 0 = 4
R <sub>i</sub> <sup>1</sup>	8 + 0 = 8	9 + 4 = 13	17 + 4 + 3 = 24	5 + 4 + 3 + 15 = 27	4 + 4 + 3 + 15 + 3 = 29
R <sub>i</sub> <sup>2</sup>		9 + 4 = 13	17 + 4 + 3 = 24	5 + 8 + 3 + 15 = 31	4 + 8 + 3 + 15 + 3 = 33
R <sub>i</sub> <sup>3</sup>				5 + 8 + 6 + 15 = 34	4 + 8 + 6 + 15 + 3 = 36
R <sub>i</sub> <sup>4</sup>				5 + 8 + 6 + 15 = 34	4 + 8 + 6 + 30 + 3 = 51
R <sub>i</sub> <sup>5</sup>					4 + 12 + 6 + 30 + 6 = 58
R <sub>i</sub> <sup>6</sup>					4 + 12 + 6 + 30 + 6 = 58

	R <sub>i</sub> [t.u.]	D <sub>i</sub> [t.u.]	R <sub>i</sub> ≤ D <sub>i</sub>
P <sub>1</sub>	8	25	sì
P <sub>2</sub>	13	30	sì
P <sub>3</sub>	24	35	sì
P <sub>4</sub>	34	40	sì
P <sub>5</sub>	58	45	no

esito positivo

sì	no
	×

risposta conclusiva

sì	no
×	

2. Nell'ipotesi che (a) i processi siano schedulati in accordo alla strategia EDF, (b) gli accessi alle risorse condivise siano gestiti secondo il protocollo stack resource policy (SRP), (c) il numero di unità disponibili per ogni tipologia di risorsa sia rispettivamente  $u_1 = 2$ ,  $u_2 = 2$ ,  $u_3 = 1$ ,  $u_4 = 1$ , (d) il numero di unità richiesto da ciascun processo per ogni tipologia di risorsa sia

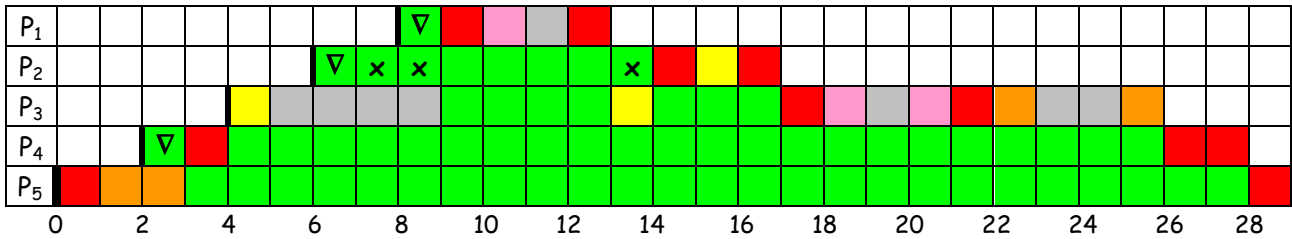
	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>	R <sub>4</sub>
P <sub>1</sub>		1		1
P <sub>2</sub>	1			
P <sub>3</sub>	2	2	1	1
P <sub>4</sub>				
P <sub>5</sub>			1	

➤ si individui il livello di preemption dei processi ed il tetto di preemption che contraddistingue ciascuna risorsa al variare del numero di unità contestualmente libere

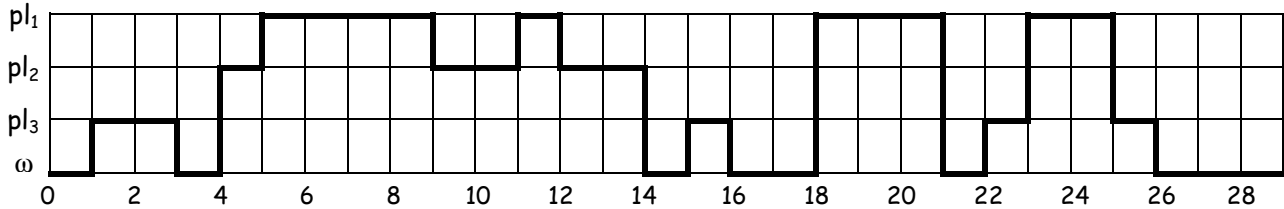
P <sub>1</sub>	pl <sub>1</sub> (max)
P <sub>2</sub>	pl <sub>2</sub>
P <sub>3</sub>	pl <sub>3</sub>
P <sub>4</sub>	pl <sub>4</sub>
P <sub>5</sub>	pl <sub>5</sub> (min)

	v <sub>k</sub> =0	v <sub>k</sub> =1	v <sub>k</sub> =2
R <sub>1</sub>	pl <sub>2</sub>	pl <sub>3</sub>	ω
R <sub>2</sub>	pl <sub>1</sub>	pl <sub>3</sub>	ω
R <sub>3</sub>	pl <sub>3</sub>	ω	-
R <sub>4</sub>	pl <sub>1</sub>	ω	-

➤ si identifichi con l'ausilio di un diagramma temporale il tempo di completamento dell'esecuzione del primo job di ciascun processo



ΠΛ<sub>5</sub>



➤ si determini il massimo tempo di blocco di ogni processo

	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>	R <sub>4</sub>	B <sub>i</sub>
P <sub>1</sub>		1		1	6
P <sub>2</sub>	1				6
P <sub>3</sub>	6	3	4	4	2
P <sub>4</sub>					2
P <sub>5</sub>			2		0

➤ si verifichi se l'insieme dei processi è schedulabile

P <sub>1</sub> : C <sub>1</sub> / D <sub>1</sub> + B <sub>1</sub> / D <sub>1</sub> = 0.4 ≤ 1	OK
P <sub>2</sub> : C <sub>1</sub> / D <sub>1</sub> + C <sub>2</sub> / D <sub>2</sub> + B <sub>2</sub> / D <sub>2</sub> = 0.46 ≤ 1	OK
P <sub>3</sub> : C <sub>1</sub> / D <sub>1</sub> + C <sub>2</sub> / D <sub>2</sub> + C <sub>3</sub> / D <sub>3</sub> + B <sub>3</sub> / D <sub>3</sub> = 0.75 ≤ 1	OK
P <sub>4</sub> : C <sub>1</sub> / D <sub>1</sub> + C <sub>2</sub> / D <sub>2</sub> + C <sub>3</sub> / D <sub>3</sub> + C <sub>4</sub> / D <sub>4</sub> + B <sub>4</sub> / D <sub>4</sub> = 0.81 ≤ 1	OK
P <sub>5</sub> : C <sub>1</sub> / D <sub>1</sub> + C <sub>2</sub> / D <sub>2</sub> + C <sub>3</sub> / D <sub>3</sub> + C <sub>4</sub> / D <sub>4</sub> + C <sub>5</sub> / D <sub>5</sub> = 0.85 ≤ 1	OK

esito positivo	
sì	no
×	

risposta conclusiva	
sì	no
×	