

PROVA SCRITTA DI SISTEMI IN TEMPO REALE (6/7/2017)

PROBLEMA n. 1

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

	T_i [t.u.]	C_i [t.u.]
P_1	10	2
P_2	15	4
P_3	20	5
P_4	30	3
P_5	60	2

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

$$M = \text{mcm}(T_1, T_2, T_3, T_4, T_5) = 60$$

- (b) dimensionamento del ciclo minore

$$\max(C_1, C_2, C_3, C_4, C_5) = 5 \leq m \leq 10 = \min(T_1, T_2, T_3, T_4, T_5) \rightarrow m = 5 \dots 10$$

$$M \bmod m = 0 \rightarrow m = 5, 6, 10$$

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

$$m = 6, i = 1, 2, 3: 2m - \text{mcd}(m, \{T_1, T_2, T_3\}) = 12 - \{2, 3, 2\} = \{10, 9, 10\} \leq \{T_1, T_2, T_3\} \rightarrow \text{OK}$$

$$m = 5, \rightarrow \text{OK}$$

$$m = 10$$

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

$$n_{cm} = 6, n_{J1} = 6, n_{J2} = 4, n_{J3} = 3, n_{J4} = 2, n_{J5} = 1$$

- (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	J ₄₂	J ₅₁
c ₁	×						×				×			×		×
c ₂		×									×			×		×
c ₃			×					×				×		×		×
c ₄				×					×			×			×	×
c ₅					×								×		×	×
c ₆						×				×			×		×	×

- (e) pianificazione dell'esecuzione di ciascun job

c ₁	J ₁₁	J ₂₁	J ₅₁		
c ₂	J ₁₂	J ₃₁	J ₄₁		
c ₃	J ₁₃	J ₂₂			
c ₄	J ₁₄	J ₂₃			
c ₅	J ₁₅	J ₃₃	J ₄₂		
c ₆	J ₁₆	J ₂₄			

esito positivo per tutti i job

sì	no	quali job ?
	×	J ₃₂

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

$$P_3 (5, 20) \Rightarrow P_3' (3, 20), P_3'' (2, 20) \text{ con } P_3' < P_3''$$

e iterazione del procedimento

c ₁	J ₁₁	J ₂₁	J ₅₁		
c ₂	J ₁₂	J ₃₁	J ₄₁		
c ₃	J ₁₃	J ₂₂	J _{32'}		
c ₄	J ₁₄	J ₂₃	J _{32''}		
c ₅	J ₁₅	J ₃₃	J ₄₂		
c ₆	J ₁₆	J ₂₄			

B) Schedulazione "priority-driven"

- Si verifichi se l'insieme di processi è schedulabile con l'algoritmo RMPO tramite:

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

S'	T _i ' [t.u.]	C _i ' [t.u.]	U _i '
P ₁ ' ≡ {P ₁ , P ₃ , P ₅ }	10	4.83	0.483
P ₂ ' ≡ {P ₂ , P ₄ }	15	5.5	0.367

$$U = U_1' + U_2' = 0.85 > U_{RMPO} (N=2) = 0.828$$

$$(1 + U_1') * (1 + U_2') = 2.03$$

esito positivo	
sì	no
	×

risposta conclusiva	
sì	no
	×

- il test di Han

S'	T _i ' [t.u.]	C _i [t.u.]	U _i '
P ₁ '	7.5	2	0.267
P ₂ ' ≡ P ₂	15	4	0.267
P ₃ '	15	5	0.333
P ₄ ' ≡ P ₄	30	3	0.100
P ₅ ' ≡ P ₅	60	2	0.033

$$U = U_1' + U_2' + U_3' + U_4' + U_5' = 1$$

esito positivo	
sì	no
×	

risposta conclusiva	
sì	no
×	

2. Nell'ipotesi che P₃, P₄ e P₅ abbiano rispettivamente deadline ([t.u.]) D₃ = 9, D₄ = 24 e D₅ = 40, si verifichi tramite il test basato sul fattore di utilizzazione efficace del processore se l'insieme di processi è schedabile con l'algoritmo DMPO

- il test basato sul fattore di utilizzazione efficace del processore

	H _n	H ₁	f	U(n, δ)	f ≤ U(n, δ)
P ₃	{ }	{ }	5/20=0.25	U(1,0.45)=0.45	sì
P ₁	{ }	{P ₃ }	(5+2)/10=0.7	U(1,1)=1	sì
P ₂	{P ₁ }	{P ₃ }	2/10+(5+4)/15=0.8	U(2,1)=0.828	sì
P ₄	{ P ₃ , P ₁ , P ₂ }	{ }	5/20+2/10+4/15+3/30=0.817	U(4,0.8)=0.699	no
P ₅	{ P ₃ , P ₁ , P ₂ , P ₄ }	{ }	5/20+2/10+4/15+3/30+2/60=0.85	U(5,0.667)=0.629	no

esito positivo	
sì	no
	×

risposta conclusiva	
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'approccio "processor demand":

$$t^* = ((1-9/20)*5+(1-24/30)*3+(1-40/60)*2)/(1-0.85) = 26.8$$

n	BI ⁿ
0	2 + 4 + 5 + 3 + 2 = 16
1	4 + 8 + 5 + 3 + 2 = 22
2	6 + 8 + 10 + 3 + 2 = 29
3	6 + 8 + 10 + 3 + 2 = 29

$$BI = 29$$

$$\min \{t^*, BI\} = 26.8$$

$$\mathcal{D}^* \cap \mathcal{D} = \{9, 10, 15, 20, 24\}$$

t	C ₁ (0,t)	C ₂ (0,t)	C ₃ (0,t)	C ₄ (0,t)	C ₅ (0,t)	C _p (0,t)	≤ t
9	0	0	5	0	0	5	sì
10	2	0	5	0	0	7	sì
15	2	4	5	0	0	11	sì
20	4	4	5	0	0	13	sì
24	4	4	5	3	0	16	sì

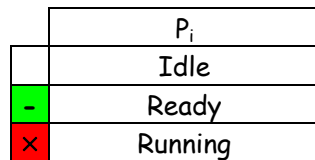
esito positivo

sì	no
×	

risposta conclusiva

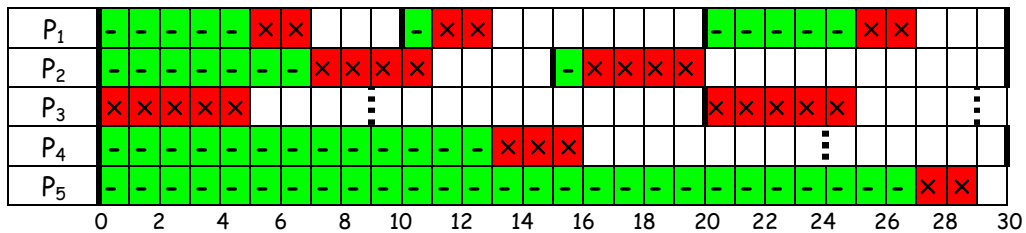
sì	no
×	

4. Nelle stesse ipotesi di cui al punto precedente e facendo riferimento alla seguente notazione



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
P ₁	8	8	-	8
P ₂	11	4	11	-
P ₃	4	-	-	4
P ₄	21	11	8	-
P ₅	38	28	23	18



esito positivo

sì	no
×	

risposta conclusiva

sì	no
×	

PROBLEMA n. 2

Un sistema di elaborazione monoprocesso ha il compito di eseguire tre processi periodici P_1, P_2, P_3 tra loro indipendenti, gestendo nel contempo cinque richieste aperiodiche $R_{a1}, R_{a2}, R_{a3}, R_{a4}, R_{a5}$:

	T_i [t.u.]	C_i [t.u.]
P_1	10	3
P_2	15	3
P_3	35	7

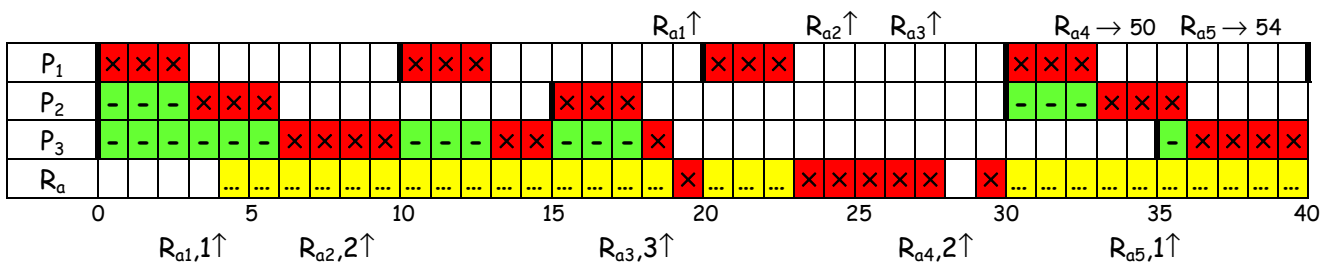
	a_i [t.u.]	C_i [t.u.]
R_{a1}	4	1
R_{a2}	9	2
R_{a3}	19	3
R_{a4}	29	2
R_{a5}	36	1

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

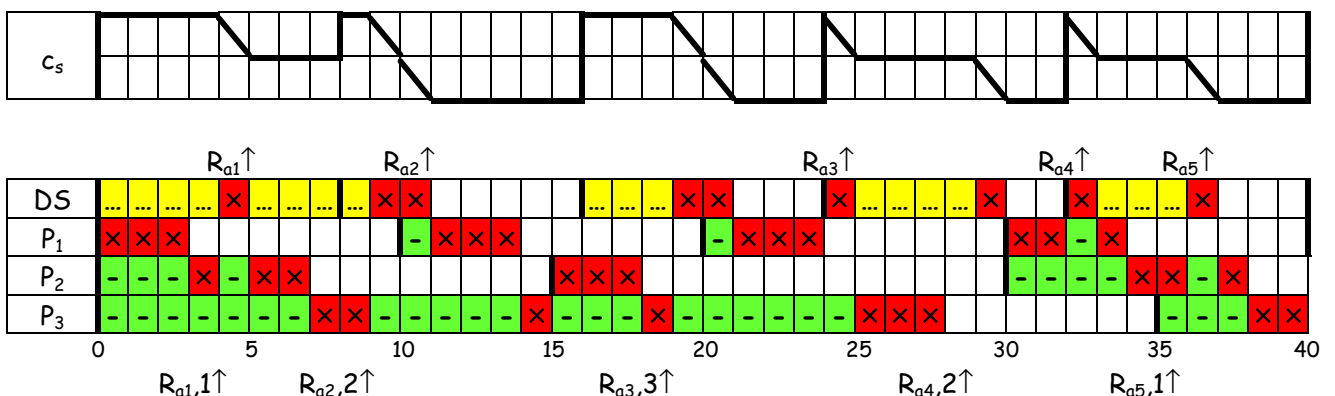
si determini il tempo di completamento dell'esecuzione delle richieste aperiodiche derivanti dalla loro gestione tramite

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

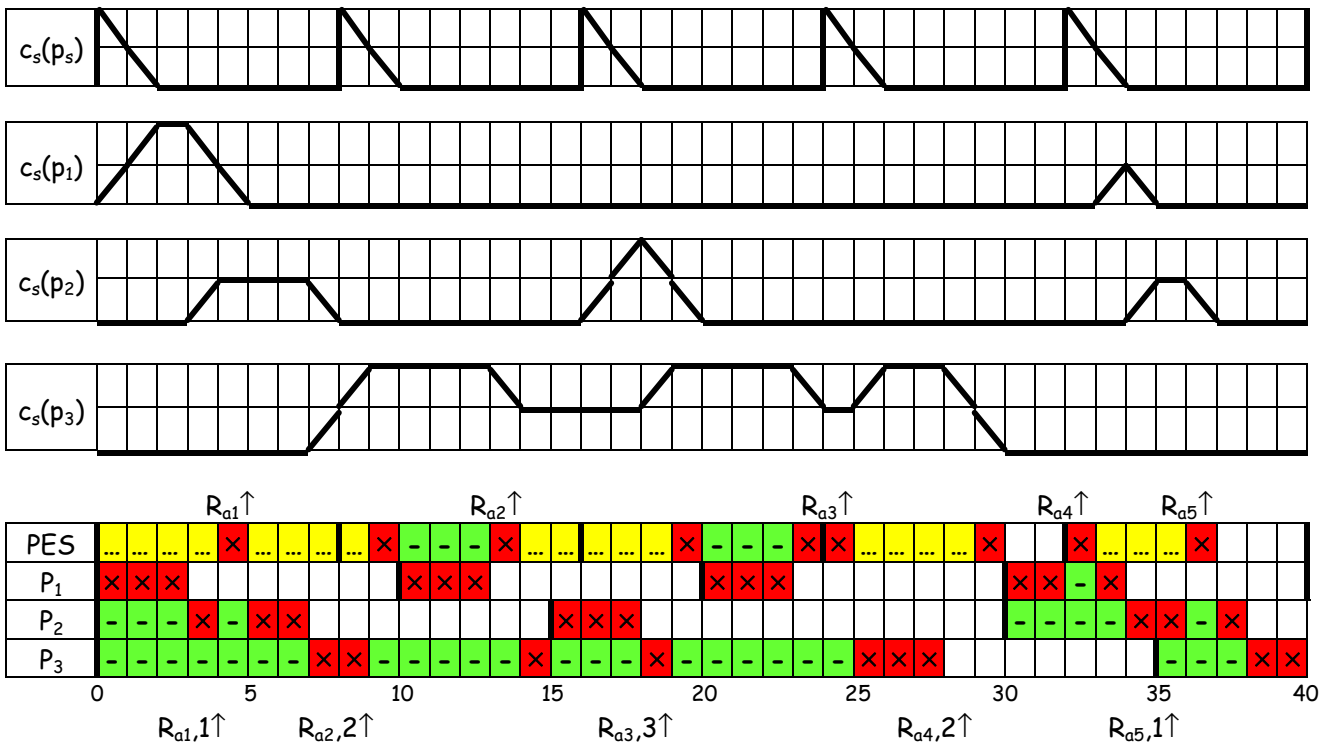


2. Server a priorità statica di periodo $T_s = 8$ t.u. e capacità $C_s = 2$ t.u. (strategia di schedulazione dei processi e del Server: RMPO)

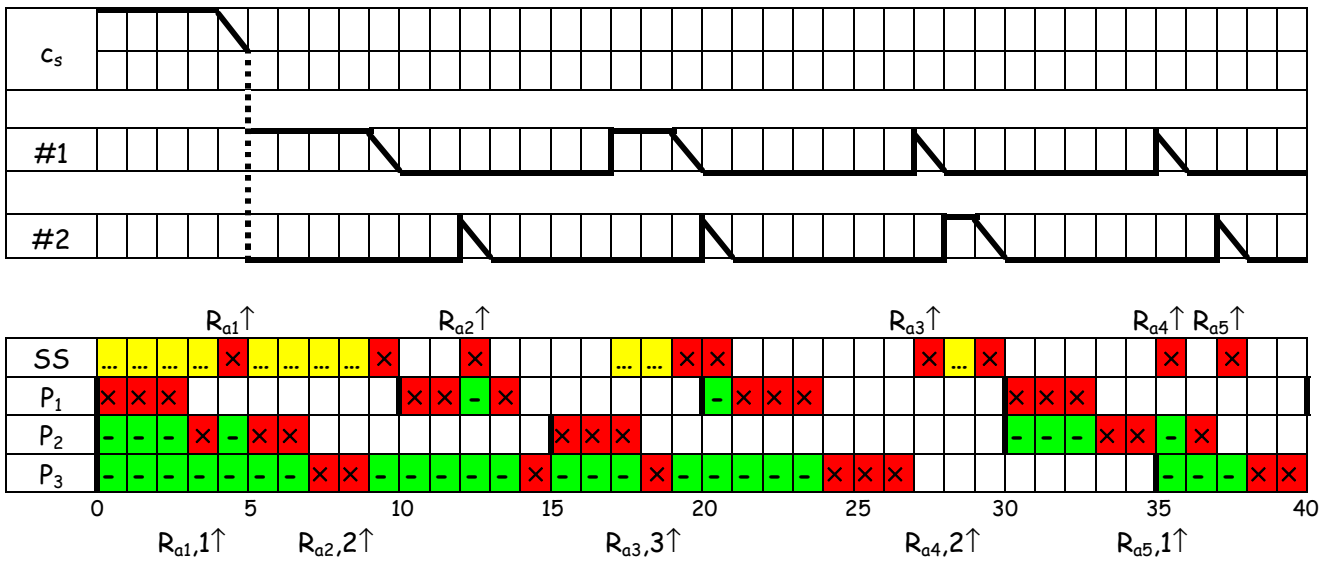
• Deferrable Server



• Priority Exchange Server



• Sporadic Server



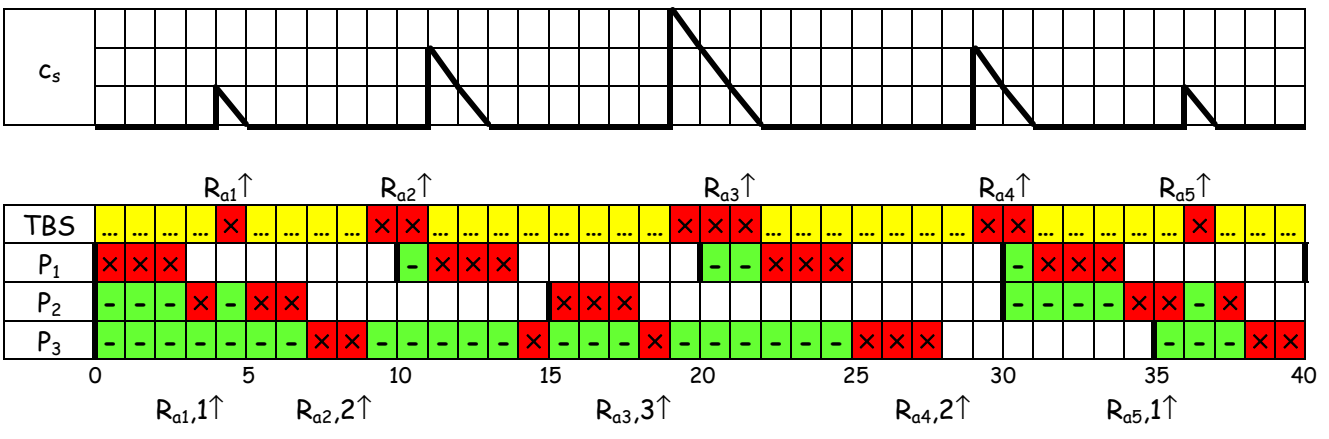
		#1	#2	#1	#2	#1	#2	#1	#2
t_E	4	9	12	19	20	27	29	35	37
t_D	5	10	13	20	21	28	30	36	38
RA	1	1	1	1	1	1	1	1	1
RT	12	17	20	27	28	35	37	43	45

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_{s_i} [t.u.]
R_{a1}	4	1	$4 + 1 / 0.3 = 7,3$
R_{a2}	9	2	$9 + 2 / 0.3 = 15,7$
R_{a3}	19	3	$19 + 3 / 0.3 = 29,0$
R_{a4}	29	2	$29 + 2 / 0.3 = 35,7$
R_{a5}	36	1	$36 + 1 / 0.3 = 39,3$



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	20	20	6
P_2	25	25	5
P_3	30	30	5
P_4	45	40	5
P_5	60	60	5

attivati rispettivamente all'istante ([t.u.]) $\phi_1 = 8$, $\phi_2 = 6$, $\phi_3 = 4$, $\phi_4 = 2$, $\phi_5 = 0$, è affidata ad un sistema di elaborazione monoproiettore.

I processi 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_1 (4 t.u. complessivamente) con accesso annidato dopo 1 t.u. a R_2 (2 t.u.), esecuzione normale (1 t.u.);
- P_2 : esecuzione normale (1 t.u.), accesso a R_1 (2 t.u.), esecuzione normale (2 t.u.);
- P_3 : esecuzione normale (1 t.u.), accesso a R_2 (3 t.u. complessivamente) con accesso annidato dopo 1 t.u. a R_3 (1 t.u.), esecuzione normale (1 t.u.);
- P_4 : esecuzione normale (1 t.u.), accesso a R_3 (3 t.u. complessivamente) con accesso annidato dopo 1 t.u. a R_4 (1 t.u.), esecuzione normale (1 t.u.);
- P_5 : esecuzione normale (1 t.u.), accesso a R_4 (3 t.u.), esecuzione normale (1 t.u.).

	C_{type}					
P_1		R ₁	R ₂	R ₂	R ₁	
P_2		R ₁	R ₁			
P_3		R ₂	R ₃	R ₂		
P_4		R ₃	R ₄	R ₃		
P_5		R ₄	R ₄	R ₄		

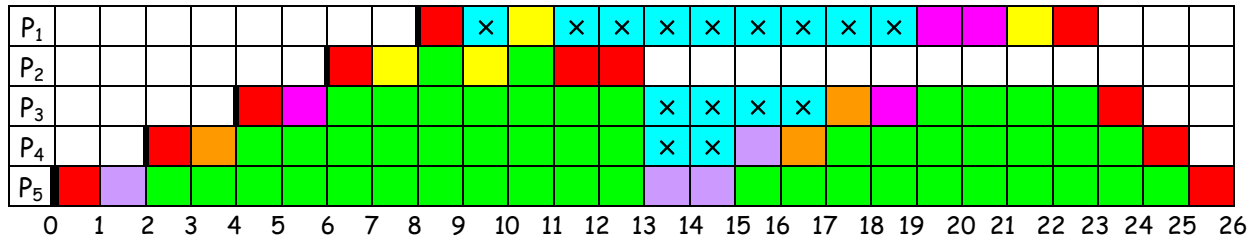
1. Nell'ipotesi che i processi siano schedulati in accordo alla strategia DMPO,

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

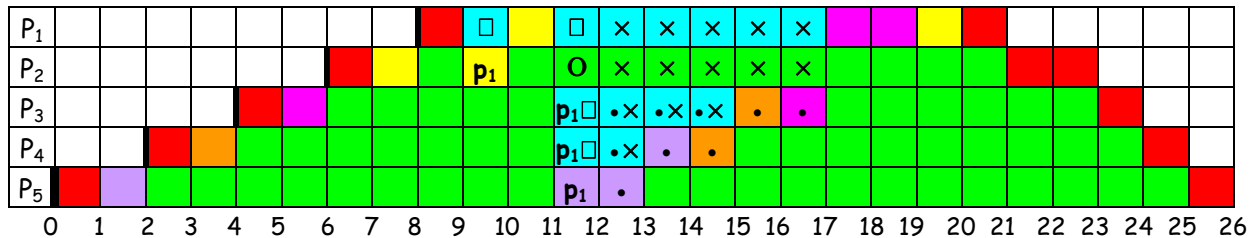
	P_i
	Idle
	ready ($\times [O, \nabla]$ when blocked)
	suspended ($\times [\square, \nabla]$ when blocked)
	running without locking any semaphore
	executing critical section protected by semaphore S_1
	executing critical section protected by semaphore S_2
	executing critical section protected by semaphore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

ed evidenziando esplicitamente la priorità corrente (π_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:

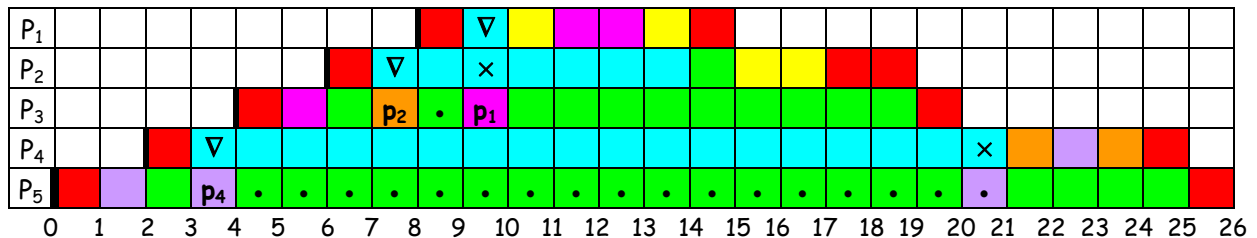
- esecuzione in base alle relative priorità statiche (NOP)



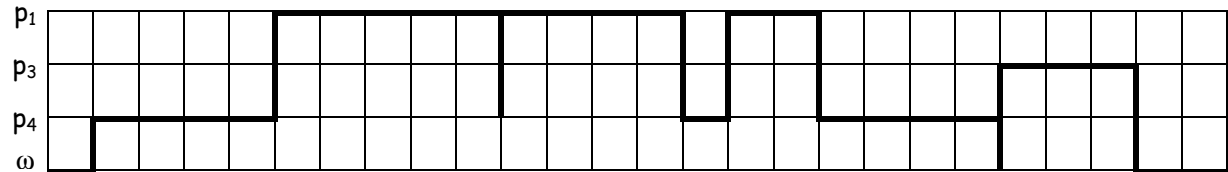
- applicazione del protocollo priority inheritance (PIP)



- applicazione del protocollo priority ceiling (PCP) [$PC_1 = p_1, PC_2 = p_1, PC_3 = p_3, PC_4 = p_4$]



$\Pi \Gamma_s$



	ϕ_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	6	23	9	21	7	15	1
P_2	6	5	13	2	23	12	19	8
P_3	4	5	24	15	24	15	20	11
P_4	2	5	25	18	25	18	25	18
P_5	0	5	26	21	26	21	26	21
			NOP		PIP		PCP	

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

	R ₁	R ₂	R ₃	R ₄	B _i	
P ₁	4	2			11	3
P ₂	2				9	3
P ₃		3	1		6	3
P ₄			3	1	3	3
P ₅				3	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 ₁	P ₂	P ₃	P ₄	P ₅
R _i ⁰	6 + 11 = 17	5 + 9 = 14	5 + 6 = 11	5 + 3 = 8	5 + 0 = 5
R _i ¹	17 + 0 = 17	14 + 6 = 20	11 + 6 + 5 = 22	8 + 6 + 5 + 5 = 24	5 + 6 + 5 + 5 + 5 = 26
R _i ²		14 + 6 = 20	11 + 12 + 5 = 28	8 + 12 + 5 + 5 = 30	5 + 12 + 10 + 5 + 5 = 37
R _i ³			11 + 12 + 10 = 33	8 + 12 + 10 + 5 = 35	5 + 12 + 10 + 10 + 5 = 42
R _i ⁴			11 + 12 + 10 = 33	8 + 12 + 10 + 10 = 40	5 + 18 + 10 + 10 + 5 = 48
R _i ⁵				8 + 12 + 10 + 10 = 40	5 + 18 + 10 + 10 + 10 = 53
R _i ⁶					5 + 18 + 15 + 10 + 10 = 58
R _i ⁷					5 + 18 + 15 + 10 + 10 = 58

	T _i [t.u.]	D _i [t.u.]	C _i [t.u.]	R _i [t.u.]	R _i ≤ D _i
P ₁	20	20	6	17	sì
P ₂	25	25	5	20	sì
P ₃	30	30	5	33	no
P ₄	45	40	5	40	sì
P ₅	60	60	5	58	sì

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 $u_1 = 2$, $u_2 = 1$, $u_3 = 1$, $u_4 = 2$, (d) il numero di unità richiesto da ciascun processo per ogni tipologia di risorsa sia

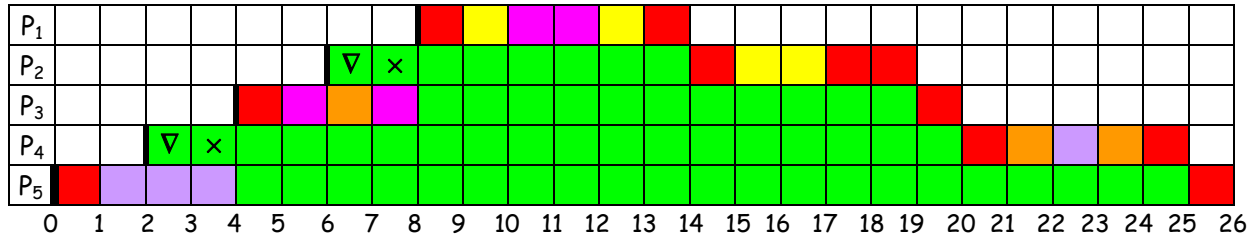
	R ₁	R ₂	R ₃	R ₄
P ₁	1	1		
P ₂	2			
P ₃		1	1	
P ₄			1	1
P ₅				2

- 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

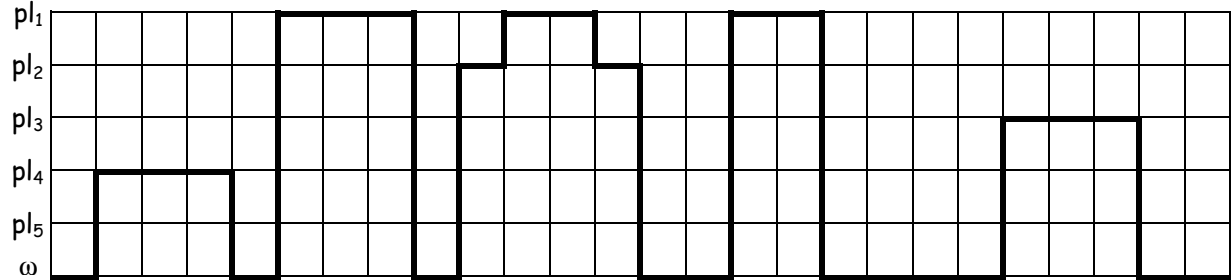
P ₁	pl ₁
P ₂	pl ₂
P ₃	pl ₃
P ₄	pl ₄
P ₅	pl ₅

	v _k =0	v _k =1	v _k =2
R ₁	pl ₁	pl ₂	ω
R ₂	pl ₁	ω	-
R ₃	pl ₃	ω	-
R ₄	pl ₄	pl ₅	ω

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



ΠΛ_s



➤ si determini il massimo tempo di blocco di ogni processo

	R ₁	R ₂	R ₃	R ₄	B _i
P ₁	4	2			3
P ₂	2				3
P ₃		3	1		3
P ₄			3	1	3
P ₅				3	0

➤ si verifichi se l'insieme dei processi è schedabile

P₁ : C₁ / D₁ + B₁ / D₁ = 0.45 ≤ 1 OK

P₂ : C₁ / D₁ + C₂ / D₂ + B₂ / D₂ = 0.62 ≤ 1 OK

P₃ : C₁ / D₁ + C₂ / D₂ + C₃ / D₃ + B₃ / D₃ = 0.767 ≤ 1 OK

P₄ : C₁ / D₁ + C₂ / D₂ + C₃ / D₃ + C₄ / D₄ + B₄ / D₄ = 0.867 ≤ 1 OK

P₅ : C₁ / D₁ + C₂ / D₂ + C₃ / D₃ + C₄ / D₄ + C₅ / D₅ = 0.875 ≤ 1 OK

esito positivo	
sì	no
×	

risposta conclusiva	
sì	no
×	