

Sistemi Embedded e Real-time (M. Cesati)

Esempio di prova scritta, 25 giugno 2009

Esercizio 1. Il seguente sistema di task periodici è schedulato su un processore con un algoritmo “cyclic schedule”: $T_1 = (6, 1)$, $T_2 = (10, 2)$, $T_3 = (18, 2)$.

(a) Determinare una dimensione appropriata per il frame. I job sono interrompibili, ma il numero di interruzioni deve essere il più piccolo possibile.

(b) Determinare una schedulazione dei task periodici nell’intervallo $[0, 30]$, utilizzando la dimensione del frame determinata in (a).

Esercizio 2. Si consideri un sistema di task periodici, indipendenti e interrompibili schedulati su un processore: $T_1 = (10, 4)$, $T_2 = (20, 4)$, $T_3 = (40, 4)$, $T_4 = (40, 3)$.

(a) Determinare analiticamente se il sistema è schedulabile con EDF.

(b) Determinare analiticamente se il sistema è schedulabile con RM.

(c) Al sistema di task viene aggiunto $T_5 = (30, 1)$. Determinare analiticamente se il nuovo sistema è schedulabile con RM.

Esercizio 3. Un sistema deve eseguire cinque job J_1, \dots, J_5 , con J_i di priorità maggiore di J_k se $i < k$. I job utilizzano tre risorse condivise R_1, R_2 e R_3 , e le relative sezioni critiche sono: $J_1 : [R_1; 2]$, $J_2 : \text{nessuna}$, $J_3 : [R_2; 1]$, $J_4 : [R_1; 3 [R_3; 1]]$, $J_5 : [R_2; 4 [R_3; 2]]$.

(a) Quali sono i tempi massimi di blocco dei job per conflitto di risorse utilizzando il protocollo NPCS?

(b) Quali sono i tempi massimi di blocco dei job per conflitto di risorse utilizzando il protocollo priority-ceiling?

Sistemi Embedded e Real-time (M. Cesati)

Esempio di soluzioni della prova scritta, 25 giugno 2009

Esercizio 1. *Il seguente sistema di task periodici è schedulato su un processore con un algoritmo “cyclic schedule”: $T_1 = (6, 1)$, $T_2 = (10, 2)$, $T_3 = (18, 2)$.*

(a) *Determinare una dimensione appropriata per il frame. I job sono interrompibili, ma il numero di interruzioni deve essere il più piccolo possibile.*

- Vincolo sui tempi d'esecuzione dei job:

$$f \geq \max\{1, 2, 2\} = 2$$

- Vincolo sulla divisibilità della lunghezza dell'iperperiodo ($\text{lcm}\{6, 10, 18\} = 90$):

$$f \in \{2, 3, 5, 6, 9, 10, 15, 18, 30, 45, 90\}$$

- Vincolo sul task T_1 ($2f - \text{gcd}\{f, 6\} \leq 6$):

$$\begin{aligned} 2 \cdot 2 - \text{gcd}\{6, 2\} &= 2 \leq 6 \\ 2 \cdot 3 - \text{gcd}\{6, 3\} &= 3 \leq 6 \\ 2 \cdot 5 - \text{gcd}\{6, 5\} &= 9 > 6 && \Rightarrow f \neq 5 \\ 2 \cdot 6 - \text{gcd}\{6, 6\} &= 6 \leq 6 \\ 2 \cdot x - \text{gcd}\{6, x\} &> 6 \text{ if } x > 6 && \Rightarrow f \leq 6 \end{aligned}$$

- Vincolo sul task T_2 ($2f - \text{gcd}\{f, 10\} \leq 10$):

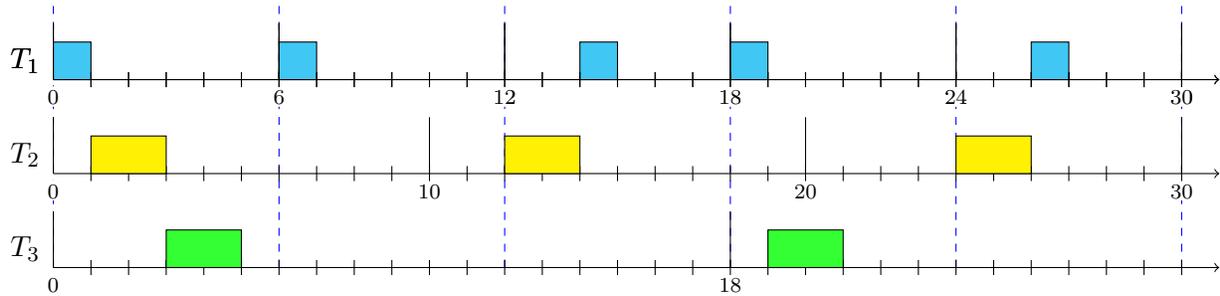
$$\begin{aligned} 2 \cdot 2 - \text{gcd}\{10, 2\} &= 2 \leq 10 \\ 2 \cdot 3 - \text{gcd}\{10, 3\} &= 5 \leq 10 \\ 2 \cdot 6 - \text{gcd}\{10, 6\} &= 10 \leq 10 \end{aligned}$$

- Vincolo sul task T_3 ($2f - \text{gcd}\{f, 18\} \leq 18$):

$$\begin{aligned} 2 \cdot 2 - \text{gcd}\{18, 2\} &= 2 \leq 18 \\ 2 \cdot 3 - \text{gcd}\{18, 3\} &= 3 \leq 18 \\ 2 \cdot 6 - \text{gcd}\{18, 6\} &= 6 \leq 18 \end{aligned}$$

Considerando le varie condizioni, $f \in \{2, 3, 6\}$. Il valore $f = 6$ minimizza l'overhead dovuto alle esecuzioni dello scheduler, ed è quindi probabilmente la scelta migliore.

(b) Determinare una schedulazione dei task periodici nell'intervallo $[0, 30]$ utilizzando la dimensione del frame determinata in (a).



Equivalentemente, la schedulazione è descrivibile elencando i blocchi di schedulazione:

- Frame 0: $L(0) = T_1, T_2, T_3$
- Frame 1: $L(1) = T_1$
- Frame 2: $L(2) = T_2, T_1$
- Frame 3: $L(3) = T_1, T_3$
- Frame 4: $L(4) = T_2, T_1$

Esercizio 2. Si consideri un sistema di task periodici, indipendenti e interrompibili schedulati su un processore: $T_1 = (10, 4)$, $T_2 = (20, 4)$, $T_3 = (40, 4)$, $T_4 = (40, 3)$.

(a) Determinare analiticamente se il sistema è schedulabile con EDF.

Confrontiamo il fattore di utilizzazione del sistema con quello di EDF:

$$U = \frac{4}{10} + \frac{4}{20} + \frac{4}{40} + \frac{3}{40} = \frac{31}{40} < 1 = U_{\text{UDF}}$$

Di conseguenza, il sistema di task è schedulabile con EDF.

(b) Determinare analiticamente se il sistema è schedulabile con RM.

Si osservi che i task sono semplicemente periodici, in quanto per ogni coppia di periodi uno di essi è un multiplo intero dell'altro.

In tali condizioni, il fattore di utilizzazione per RM coincide con quello di EDF, ossia è 1. Perciò il sistema di task è schedulabile con RM.

(c) Al sistema di task viene aggiunto $T_5 = (30, 1)$. Determinare analiticamente se il nuovo sistema è schedulabile con RM.

Il task T_5 rende il sistema non più semplicemente periodico. Il fattore di utilizzazione diventa

$$U = \frac{31}{40} + \frac{1}{30} = \frac{97}{120}$$

Poiché è possibile partizionare i task in due sottoinsiemi, ciascuno dei quali è semplicemente periodico, per il teorema di Kuo-Mok la condizione di schedulabilità diventa:

$$U_{\text{RM}}(2) = 2 \cdot (2^{1/2} - 1) > 0.828 > 0.808 > \frac{97}{120} = U$$

Di conseguenza si conclude che il nuovo sistema di task è schedulabile.

A titolo di esempio, svolgiamo l'esercizio anche senza considerare il Teorema di Kuo-Mok; consideriamo perciò l'insieme di task nel suo complesso.

- Condizione di Liu-Layland: $U_{\text{RM}}(5) = 5 \cdot (2^{1/5} - 1) < 0.743 < 0.808 < \frac{97}{120} = U$
- Test iperbolico: $\left(1 + \frac{4}{10}\right) \cdot \left(1 + \frac{4}{20}\right) \cdot \left(1 + \frac{4}{40}\right) \cdot \left(1 + \frac{3}{40}\right) \cdot \left(1 + \frac{1}{30}\right) = \frac{102\,641}{50\,000} > 2$
- Applichiamo il test di schedulabilità calcolando la funzione di tempo necessario

$$w_i(t) = e_i + \sum_k \left\lceil \frac{t}{p_k} \right\rceil \cdot e_k$$

ove la sommatoria è estesa a tutti i task con priorità maggiore a quella del task in esame.

- Test di schedulabilità per T_1 : $w_1(t) = 4 \leq 10 \quad \Rightarrow$ schedulabile.
- Test di schedulabilità per T_2 : $w_2(t) = 4 + \lceil t/10 \rceil \cdot 4$;
 $w_2(10) = 8 \leq 10 \quad \Rightarrow$ schedulabile

(Notare che i risultati dei test di schedulabilità per i task di priorità maggiore del nuovo task, ossia per T_1 e T_2 , erano ovvi a causa della soluzione data al punto (b) e dunque potevano essere omessi.)

- Test di schedulabilità per T_5 : $w_5(t) = 1 + \lceil t/10 \rceil \cdot 4 + \lceil t/20 \rceil \cdot 4$;
 $w_5(10) = 9 \leq 10 \quad \Rightarrow$ schedulabile
- Test di schedulabilità per T_3 : $w_3(t) = 4 + \lceil t/10 \rceil \cdot 4 + \lceil t/20 \rceil \cdot 4 + \lceil t/30 \rceil$;
 $w_3(10) = 13 > 10$
 $w_3(20) = 17 \leq 20 \quad \Rightarrow$ schedulabile
- Test di schedulabilità per T_4 : $w_4(t) = 3 + \lceil t/10 \rceil \cdot 4 + \lceil t/20 \rceil \cdot 4 + \lceil t/30 \rceil + \lceil t/40 \rceil \cdot 4$;
 $w_4(10) = 16 > 10$
 $w_4(20) = 20 \leq 20 \quad \Rightarrow$ schedulabile

- In alternativa si potevano calcolare i massimi tempi di risposta di T_5 , T_3 e T_4 risolvendo iterativamente l'equazione $w_i(t) = t$ e controllando se la soluzione è non maggiore della scadenza relativa:

- Il massimo tempo di risposta di T_5 è 9 ($w_5(1) = 9$; $w_5(9) = 9$) ed è inferiore alla scadenza relativa 30, quindi T_5 è schedulabile.
- Il massimo tempo di risposta di T_3 è 17 ($w_3(4) = 13$; $w_3(13) = 17$; $w_3(17) = 17$) ed è inferiore alla scadenza relativa 40, quindi T_3 è schedulabile.
- Il massimo tempo di risposta di T_4 è 20 ($w_4(3) = 16$; $w_4(16) = 20$; $w_4(20) = 20$) ed è inferiore alla scadenza relativa 40, quindi T_4 è schedulabile.

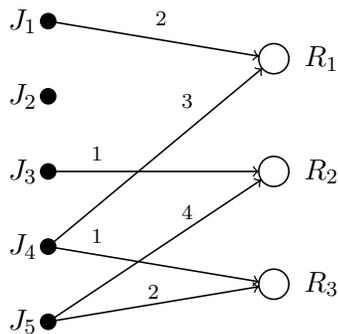
Esercizio 3. Un sistema deve eseguire cinque job J_1, \dots, J_5 , con J_i di priorità maggiore di J_k se $i < k$. I job utilizzano tre risorse condivise R_1, R_2 e R_3 , e le relative sezioni critiche sono: $J_1 : [R_1; 2]$, $J_2 : \text{nessuna}$, $J_3 : [R_2; 1]$, $J_4 : [R_1; 3 [R_3; 1]]$, $J_5 : [R_2; 4 [R_3; 2]]$.

(a) Quali sono i tempi massimi di blocco dei job per conflitto di risorse utilizzando il protocollo NPCS?

Il massimo tempo di blocco per conflitto di risorse $b_i(\text{rc})$ del job J_i è dato dalla lunghezza della più lunga sezione critica tra tutti i job di priorità inferiore. Perciò:

$$b_1(\text{rc}) = 4; \quad b_2(\text{rc}) = 4; \quad b_3(\text{rc}) = 4; \quad b_4(\text{rc}) = 4; \quad b_5(\text{rc}) = 0.$$

(b) Quali sono i tempi massimi di blocco dei job per conflitto di risorse utilizzando il protocollo priority-ceiling?



B_d	J_2	J_3	J_4	J_5
J_1			3	
J_2	*			
J_3		*		4
J_4			*	2

B_i	J_2	J_3	J_4	J_5
J_1				
J_2	*		3	
J_3		*	3	
J_4			*	4

(notare che non è necessario derivare la tabella B_c dei tempi di blocco dovuto al priority ceiling perché tutti i job hanno priorità distinte)

I tempi di blocco per conflitti di risorse di ciascun job sono determinati dal valore massimo su ciascuna riga delle tabelle:

$$b_1(\text{rc}) = 3; \quad b_2(\text{rc}) = 3; \quad b_3(\text{rc}) = 4; \quad b_4(\text{rc}) = 4; \quad b_5(\text{rc}) = 0.$$