Sistemi Embedded e Real-time (M. Cesati)

Compito scritto del 18 febbraio 2010

Esercizio 1. Il seguente sistema di task periodici è schedulato su un processore con un algoritmo "cyclic schedule": $T_1 = (10, 5, 1, 5)$, $T_2 = (4, 1, 10)$, $T_3 = (5, 2)$, $T_4 = (10, 1)$. (Si noti che il task T_1 ha fase 10, mentre gli altri task hanno fase 0.)

(a) Determinare una dimensione intera appropriata per il frame. I job non sono interrompibili.

(b) Determinare una schedulazione ciclica valida per il sistema utilizzando la dimensione del frame determinata in (a), ovvero dimostrare formalmente che non esiste una tale schedulazione.

Esercizio 2. Un server procastinabile con periodo $p_s = 4$, budget $e_s = 2$ e fase indeterminata è schedulato su un singolo processore insieme a due task periodici, indipendenti e interrompibili: $T_1 = (10, 1)$ e $T_2 = (11, 2)$.

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

(b) Indicare un esempio di schedulazione RM in cui il task T_2 presenta il peggiore tempo di risposta possibile. In quali condizioni ciò si verifica?

Esercizio 3. Un sistema deve eseguire cinque job $J_1, \ldots, J_5$, 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: $J_1 : [R_1; 4]$, J_2 : nessuna, $J_3 : [R_2; 3]$, $J_4 : [R_1; 3]\,[R_3; 1]$, $J_5 : [R_2; 2]\,[R_3; 1]$.

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

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

Sistemi Embedded e Real-time (M. Cesati)

Soluzioni del compito scritto del 18 febbraio 2010

Esercizio 1. *Il seguente sistema di task periodici è schedulato su un processore con un algoritmo "cyclic schedule":* $T_1 = (10, 5, 1, 5)$, $T_2 = (4, 1, 10)$, $T_3 = (5, 2)$, $T_4 = (10, 1)$. *(Si noti che il task* T_1 *ha fase 10, mentre gli altri task hanno fase 0.)*

(a) *Determinare una dimensione intera appropriata per il frame. I job non sono interrompibili.*

- Vincolo sulle fasi dei task: il primo job di ciascun task deve essere rilasciato esattamente all'inizio di un frame. In altri termini, la fase di ciascun task deve essere un multiplo intero non negativo della dimensione del frame.

  Poiché T_1 ha fase 10, dieci deve essere un multiplo intero di f. Gli altri task con fase 0 non pongono alcun vincolo.

- Vincolo sui tempi d'esecuzione dei job:

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

- Vincolo sulla divisibilità della lunghezza dell'iperperiodo ($H = \text{mcm}\{5, 4, 10\} = 20$):

$$f \in \{1, 2, 4, 5, 10, 20\}.$$

  Considerando i vincoli precedenti: $f \in \{2, 5, 10\}$.

- Vincolo sul task T_1 ($2f - \gcd\{5, f\} \leq 5$):

$$\begin{array}{lll} f = 2: & 2 \cdot 2 - \gcd\{5, 2\} = 3 \leq 5 & \\ f = 5: & 2 \cdot 5 - \gcd\{5, 5\} = 5 \leq 5 & \\ f = 10: & 2 \cdot 10 - \gcd\{5, 10\} = 15 > 5 & \Rightarrow f \neq 10 \end{array}$$

- Vincolo sul task T_2 ($2f - \gcd\{4, f\} \leq 10$):

$$\begin{array}{ll} f = 2: & 2 \cdot 2 - \gcd\{4, 2\} = 2 \leq 10 \\ f = 5: & 2 \cdot 5 - \gcd\{4, 5\} = 9 \leq 10 \end{array}$$

- Vincolo sul task T_3 ($2f - \gcd\{5, f\} \leq 5$):

$$\begin{array}{ll} f = 2: & 2 \cdot 2 - \gcd\{5, 2\} = 3 \leq 5 \\ f = 5: & 2 \cdot 5 - \gcd\{5, 5\} = 5 \leq 5 \end{array}$$

- Vincolo sul task T_4 ($2f - \gcd\{10, f\} \leq 10$):

$$\begin{aligned} f = 2: &\quad 2 \cdot 2 - \gcd\{10, 2\} = 2 \leq 10 \\ f = 5: &\quad 2 \cdot 5 - \gcd\{10, 5\} = 5 \leq 10 \end{aligned}$$

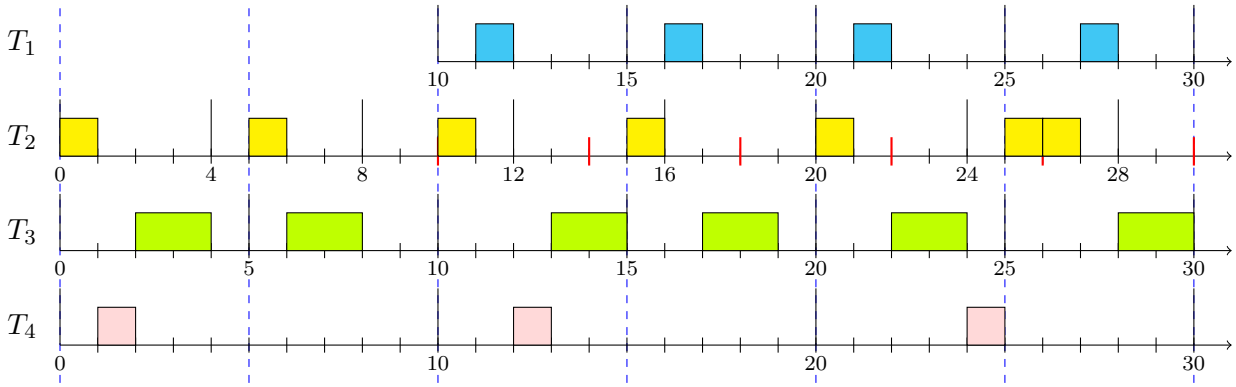
Benché sia 2 che 5 siamo ammissibili, è preferibile scegliere come dimensione del frame $f = 5$ per minimizzare il numero di invocazioni dello scheduler.

(b) *Determinare una schedulazione ciclica valida per il sistema utilizzando la dimensione del frame determinata in (a), ovvero dimostrare formalmente che non esiste una tale schedulazione.*

Verifichiamo preventivamente il fattore di utilizzazione globale del sistema di task periodici:

$$U_T = \frac{1}{5} + \frac{1}{4} + \frac{2}{5} + \frac{1}{10} = \frac{19}{20} \leq 1$$

Una schedulazione valida nell'intervallo $[0, 30]$ è la seguente:



La schedulazione nell'intervallo $[10, 30]$ può essere ripetuta all'infinito. Infatti:

- Sia all'istante 10 che all'istante 30 vengono rilasciati i job dei task T_1, T_3 e T_4; inoltre tutti i job di questi task rilasciati in precedenza sono stati completati.
- Sia all'istante $10 - 2 = 8$ che all'istante $30 - 2 = 28$ vengono rilasciati job del task T_2, e questi non sono ancora stati eseguiti agli istanti 10 e 30; inoltre tutti gli altri job di T_2 rilasciati in precedenza sono stati completati.

Risulta così determinata una schedulazione ciclica valida per il sistema, rappresentabile per mezzo dei seguenti blocchi di schedulazione ($k = 0, 1, 2, \ldots$):

$$\begin{aligned} L(0) &= \langle T_2, T_4, T_3 \rangle \\ L(1) &= \langle T_2, T_3 \rangle \\ L(4k+2) &= \langle T_2, T_1, T_4, T_3 \rangle \\ L(4k+3) &= \langle T_2, T_1, T_3 \rangle \\ L(4k+4) &= \langle T_2, T_1, T_3, T_4 \rangle \\ L(4k+5) &= \langle T_2, T_2, T_1, T_3 \rangle \end{aligned}$$

Esercizio 2. *Un server procastinabile con periodo $p_s = 4$, budget $e_s = 2$ e fase indeterminata è schedulato su un singolo processore insieme a due task periodici, indipendenti e interrompibili:* $T_1 = (10, 1)$ *e* $T_2 = (11, 2)$.

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

Poichè i periodi dei task periodici e del server non rispettano le condizioni del teorema di Lehoczky, Sha e Strosnider, è necessario controllare la schedulabilità di ciascun task separatamente.

Utilizzando il fattore di utilizzazione, condizione sufficiente per la schedulabilità di T_i è:

$$\sum_{k=1}^{i} \frac{e_k}{p_k} + u_s + \frac{e_s}{p_i} \leq U_{\text{RM}}(i+1)$$

- Server procastinabile: $u_s = 2/4 \leq 1 \quad \Rightarrow$ schedulabile
- Task T_1: $1/10 + 2/4 + 2/10 = 4/5 = 0{,}8 < 0{,}828 < U_{\text{RM}}(2) \quad \Rightarrow$ schedulabile
- Task T_2: $1/10 + 2/11 + 2/4 + 2/11 = 53/55 > 0{,}96 > 0{,}78 > U_{\text{RM}}(3)$

Il task T_2 non è necessariamente schedulabile; è necessario dunque studiare la sua funzione di tempo necessario, che in presenza di server procastinabile diventa:

$$w_i(t) = e_i + e_s + \left\lceil \frac{t - e_s}{p_s} \right\rceil e_s + \sum_{k=1}^{i-1} \left\lceil \frac{t}{p_k} \right\rceil e_k$$

Perciò $w_2(t) = 4 + 2 \cdot \lceil (t-2)/4 \rceil + \lceil t/10 \rceil$:

$w_2(t) > 4$ per $t < 4$
$w_2(4) = 7 > 4$
$w_2(8) = 9 > 8$
$w_2(10) = 9 \leq 10 \qquad \Rightarrow T_2$ è schedulabile

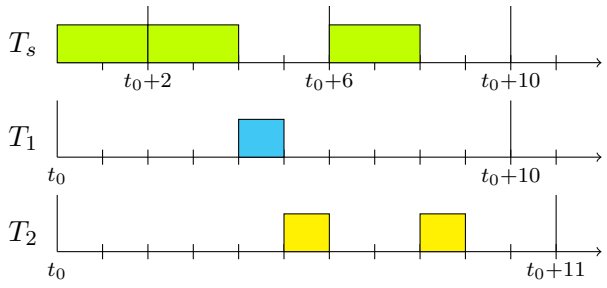
Il sistema è quindi schedulabile con RM.

(b) *Indicare un esempio di schedulazione RM in cui il task T_2 presenta il peggiore tempo di risposta possibile. In quali condizioni ciò si verifica?*

La funzione di tempo necessario $w_2(t)$ del task T_2 utilizzata nel punto (b) indica che il massimo tempo richiesto da un job di T_2 per terminare è 9 unità di tempo. Infatti, il valore 9 è la soluzione iterativa dell'equazione $w_2(t) = t$:

$$w_2(2) = 5, \quad w_2(5) = 7, \quad w_2(7) = 9, \quad w_2(9) = 9.$$

Un esempio di schedulazione RM in cui ciò si verifica è il seguente:



Le condizioni in cui tale circostanza si verifica sono:

- Al tempo t_0 vengono rilasciati job di T_1 e T_2
- Al tempo t_0 viene rilasciato un job aperiodico con tempo d'esecuzione $e \geq 6$

- Al tempo t_0 il budget del server procastinabile è al massimo, ossia è uguale a 2
- Un nuovo periodo del server procastinabile inizia a t_0+2, quindi a t_0+2 il suo budget è ripristinato

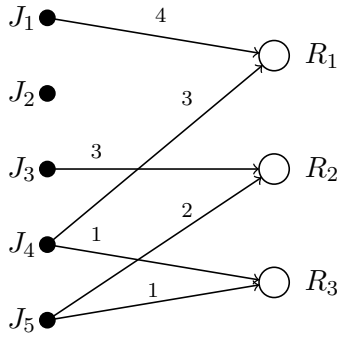
Esercizio 3. *Un sistema deve eseguire cinque job $J_1, \ldots, J_5$, con J_i di priorità maggiore di J_k se $i < k$. I job utilizzano tre risorse condivise R_1, R_2 e R_3, e le relative sezioni critiche sono: $J_1 : [R_1; 4]$, $J_2 :$ nessuna, $J_3 : [R_2; 3]$, $J_4 : [R_1; 3]\,[R_3; 1]$, $J_5 : [R_2;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

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

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

$$b_1(\mathrm{rc}) = 3; \quad b_2(\mathrm{rc}) = 3; \quad b_3(\mathrm{rc}) = 3; \quad b_4(\mathrm{rc}) = 2; \quad b_5(\mathrm{rc}) = 0.$$

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



B_d	J_2	J_3	J_4	J_5
J_1			3	
J_2	$*$			
J_3		$*$		2
J_4			$*$	1

B_i	J_2	J_3	J_4	J_5
J_1				
J_2	$*$		3	
J_3		$*$	3	
J_4			$*$	2

Assumendo che i job non si auto-sospendano, i tempi di blocco per conflitti di risorse di ciascun job sono determinati dal valore massimo su ciascuna riga delle tabelle:

$$b_1(\mathrm{rc}) = 3; \quad b_2(\mathrm{rc}) = 3; \quad b_3(\mathrm{rc}) = 3; \quad b_4(\mathrm{rc}) = 2; \quad b_5(\mathrm{rc}) = 0.$$