Sistemi Embedded e Real-time (M. Cesati)

Compito scritto del 20 febbraio 2013

Esercizio 1. Il seguente sistema di task periodici non interrompibili è schedulato su un processore con un algoritmo "cyclic schedule": $T_1 = (6, 3, 1, 3)$, $T_2 = (4, 1, 5)$, $T_3 = (5, 2)$.

Determinare una schedulazione ciclica strutturata per l'insieme di task, ovvero dimostrare che tale schedulazione non esiste.

Esercizio 2. Si consideri un sistema di task periodici indipendenti, sempre interrompibili e che non si auto-sospendono: $T_1 = (6, 1/5)$, $T_2 = (8, 3/5)$, $T_3 = (14, 4)$.

(a) Supponendo che venga utilizzato uno scheduler RM e che l'overhead dello scheduler e del cambio di contesto siano trascurabili, verificare analiticamente che il sistema è schedulabile su un singolo processore.

(b) Supponendo che lo scheduler RM sia periodico, che esso controlli la coda dei job pendenti in tempo $e_0 = 1/15$, e che il costo dell'attivazione di un job sia $CS_0 = 1/5$, determinare utilizzando la condizione di Liu-Layland i possibili valori per il periodo dello scheduler che assicurano la schedulabilità del sistema.

(c) Nelle condizioni del caso (b), determinare con il test di schedulabilità se le scadenze saranno rispettate con un periodo dello scheduler pari a 2.

Esercizio 3. Un sistema deve eseguire quattro job $J_1, \ldots, J_4$, con J_i di priorità maggiore di J_k se $i < k$. I job utilizzano quattro risorse condivise $R_1, \ldots R_4$, e le relative sezioni critiche sono: $J_1 : [R_2, 3][R_3, 4]$, $J_2 : [R_3, 2][R_2, 3]$, $J_3 : [R_1, 3\,[R_4, 1]\,]$, $J_4 : [R_2, 1][R_4, 1]$.

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

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

(c) Come cambiano i tempi massimi di blocco del precedente punto (b) assumendo che nel sistema sia presente anche un job J_1' di priorità uguale a quella di J_1 e che non utilizza alcuna risorsa condivisa?

Sistemi Embedded e Real-time (M. Cesati)

Soluzioni del compito scritto del 20 febbraio 2013

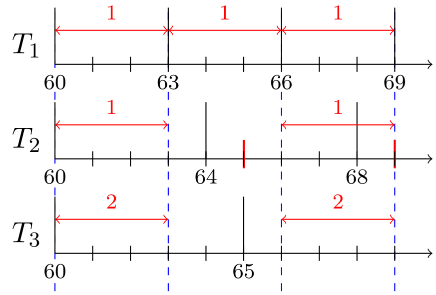
Esercizio 1. *Il seguente sistema di task periodici non interrompibili è schedulato su un processore con un algoritmo "cyclic schedule":* $T_1 = (6, 3, 1, 3)$, $T_2 = (4, 1, 5)$, $T_3 = (5, 2)$.

Determinare una schedulazione ciclica strutturata per l'insieme di task, ovvero dimostrare che tale schedulazione non esiste.

- Vincolo sulle fasi dei task: la dimensione del frame deve essere un divisore esatto di 6, quindi $f \in \{1, 2, 3, 6\}$.
- Vincolo sui tempi d'esecuzione dei job:
$$f \geq \max\{1, 1, 2\} = 2$$
- Vincolo sulla divisibilità della lunghezza dell'iperperiodo ($H = \text{mcm}\{3, 4, 5\} = 60$): tutti i valori in $\{2, 3, 6\}$ dividono esattamente l'iperperiodo.
- Vincolo sul task T_1 ($2f - \gcd\{3, f\} \leq 3$):
$$\begin{array}{ll} 2 \cdot 2 - \gcd\{3, 2\} = 3 \leq 3 & \\ 2 \cdot 3 - \gcd\{3, 3\} = 3 \leq 3 & \\ 2 \cdot 6 - \gcd\{3, 6\} = 9 > 3 & \quad \Rightarrow f \neq 6 \end{array}$$
- Vincolo sul task T_2 ($2f - \gcd\{4, f\} \leq 5$):
$$\begin{array}{l} 2 \cdot 2 - \gcd\{4, 2\} = 2 \leq 5 \\ 2 \cdot 3 - \gcd\{4, 3\} = 5 \leq 5 \end{array}$$
- Vincolo sul task T_3 ($2f - \gcd\{5, f\} \leq 5$):
$$\begin{array}{l} 2 \cdot 2 - \gcd\{5, 2\} = 3 \leq 5 \\ 2 \cdot 3 - \gcd\{5, 3\} = 5 \leq 5 \end{array}$$

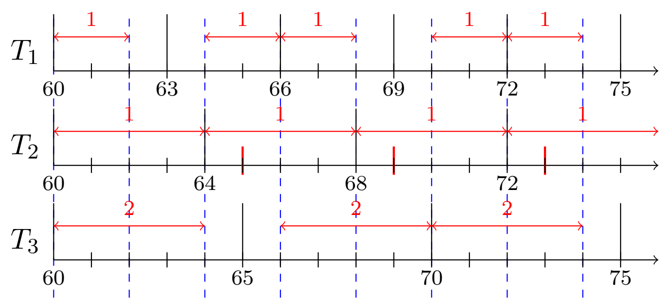
Dimensioni ammissibili per il frame sono $f = 2$ e $f = 3$.

Non esiste alcuna schedulazione ciclica strutturata per la dimensione del frame $f = 3$. Infatti poniamoci all'inizio del secondo ciclo maggiore, e supponiamo ottimisticamente che tutti i job precedentemente rilasciati siano stati completati:



Nel primo ciclo minore dovrebbero essere schedulati tutti i job dei tre task, con un tempo di esecuzione totale pari a 4, superiore alla quantità di tempo disponibile nel ciclo.

Consideriamo perciò la dimensione $f = 2$. Anche in questo caso si può dimostrare che non esiste alcuna schedulazione ciclica strutturata. Infatti, la situazione all'inizio del secondo ciclo maggiore, nel migliore dei casi, è la seguente:



Si consideri che nell'intervallo $[60, 74]$ debbono essere eseguiti almeno 5 job di T_1, 3 job di T_2 e 3 job di T_3, per un tempo d'esecuzione totale pari a 14 unità di tempo, ovvero esattamente la dimensione dell'intervallo. Quindi il processore deve essere sempre occupato in questo intervallo. Consideriamo il ciclo minore $[64, 66]$: deve essere eseguito un job di T_1 e, per non lasciare il processore inutilizzato, il job di T_2 rilasciato all'istante 64. Consideriamo ora il ciclo minore successivo $[66, 68]$: deve necessariamente essere eseguito un job di T_1, ma il rimanente slot di tempo non può essere utilizzato per un job di T_3 (in quanto non è interrompibile ed è di lunghezza 2), mentre il job di T_2 è stato già concluso nel ciclo minore precedente. Se ne conclude che il processore rimane inutilizzato per una unità di tempo in $[66, 68]$, e quindi non sarà possibile eseguire tutti i job richiesti nell'intervallo $[60, 74]$.

Esercizio 2. *Si consideri un sistema di task periodici indipendenti, sempre interrompibili e che non si auto-sospendono:* $T_1 = (6, 1/5)$, $T_2 = (8, 3/5)$, $T_3 = (14, 4)$.

(a) *Supponendo che venga utilizzato uno scheduler RM e che l'overhead dello scheduler e del cambio di contesto siano trascurabili, verificare analiticamente che il sistema è schedulabile su un singolo processore.*

Osserviamo preliminarmente che le scadenze relative dei task coincidono con i rispettivi periodi. Possiamo dunque cercare di applicare una condizione di schedulabilità basata sul fattore di utilizzazione del sistema di task.

Il fattore di utilizzazione totale del sistema è pari a

$$\frac{1}{30} + \frac{3}{40} + \frac{2}{7} = \frac{331}{840}.$$

Poiché $331/840 < 0.4 < 0.779 < U_{\mathrm{RM}}(3)$, possiamo concludere immediatamente che il sistema di task è schedulabile.

(b) *Supponendo che lo scheduler RM sia periodico, che esso controlli la coda dei job pendenti in tempo* $e_0 = 1/15$*, e che il costo dell'attivazione di un job sia* $CS_0 = 1/5$*, determinare utilizzando la condizione di Liu-Layland i possibili valori per il periodo dello scheduler che assicurano la schedulabilità del sistema.*

Poiché i job non si auto-sospendono, è possibile modellare il costo di attivazione aumentando i tempi di esecuzione dei job per un tempo pari a CS_0. Pertanto, il sistema di task equivalente (considerando anche lo scheduler T_0 con periodo incognito p_0) è

$$T_0 = (p_0, 1/15),\ \ T_1' = (6, 2/5),\ \ T_2' = (8, 4/5),\ \ T_3' = (14, 21/5).$$

La periodicità dello scheduler può rallentare l'esecuzione dei job. Infatti un job appena rilasciato non può entrare in esecuzione immediatamente, in quanto è lo scheduler che controlla la coda dei job pendenti e forza il cambio di contesto. Questo rallentamento può essere modellato considerandolo come un fattore aggiuntivo p_0 al tempo di blocco per non interrompibilità. Poiché i job per ipotesi sono sempre interrompibili si ha:

$$b_i(\mathrm{np}) = p_0, \quad \text{per } 1 \le i \le 3.$$

È necessario controllare separatamente la schedulabilità di tutti i task. Infatti ciascun task ha tempi di blocco diversi (con il caso peggiore costituito da T_1) e diversi fattori di utilizzazione da considerare (con il caso peggiore costituito da T_3).

Schedulabilità di T_1: La condizione di schedulabilità per T_1' può essere riformulata come:

$$\frac{e_1'}{p_1} + \frac{CS_0}{p_2} + \frac{CS_0}{p_3} + \frac{e_0}{p_0} + \frac{p_0}{p_1} \leq U_{\text{RM}}(4) < 0.757.$$

In questa disuguaglianza tutti i termini sono noti tranne p_0, che possiamo considerare come l'incognita x. Si ottiene pertanto la disuguaglianza

$$\frac{89}{840} + \frac{1}{15 \cdot x} + \frac{x}{6} < 0.757,$$

ossia

$$\frac{140 \cdot x^2 - 546.88 \cdot x + 56}{840 \cdot x} < 0.$$

Poiché possiamo assumere che $x > 0$, le soluzioni della disequazione sono costituite da tutti i valori entro l'intervallo determinato dalle eventuali soluzioni reali dell'equazione di secondo grado:

$$x_{1,2} = \frac{546.88 \pm \sqrt{546.88^2 - 4 \cdot 140 \cdot 56}}{2 \cdot 140} \approx \frac{546.88 \pm 517.41}{280}.$$

Pertanto la schedulabilità di T_1 è garantita per $0.11 < p_0 < 3.80$.

Schedulabilità di T_2*:* La condizione di schedulabilità per T_2' può essere riformulata come:

$$\frac{e_1'}{p_1} + \frac{e_2'}{p_2} + \frac{CS_0}{p_3} + \frac{e_0}{p_0} + \frac{p_0}{p_1} \leq U_{\text{RM}}(4) < 0.757.$$

Si ottiene pertanto la disuguaglianza

$$\frac{19}{105} + \frac{1}{15 \cdot x} + \frac{x}{8} < 0.757,$$

ossia

$$\frac{105 \cdot x^2 - 483.88 \cdot x + 56}{840 \cdot x} < 0.$$

Poiché possiamo assumere che $x > 0$, le soluzioni della disequazione sono costituite da tutti i valori entro l'intervallo determinato dalle eventuali soluzioni reali dell'equazione di secondo grado:

$$x_{1,2} = \frac{843.88 \pm \sqrt{843.88^2 - 4 \cdot 105 \cdot 56}}{2 \cdot 105} \approx \frac{483.88 \pm 458.93}{210}.$$

Pertanto la schedulabilità di T_2 è garantita per $0.12 < p_0 < 4.49$.

Schedulabilità di T_3: La condizione di schedulabilità per T_3' può essere riformulata come:

$$\frac{e_1'}{p_1} + \frac{e_2'}{p_2} + \frac{e_3'}{p_3} + \frac{e_0}{p_0} + \frac{p_0}{p_1} \leq U_{\mathrm{RM}}(4) < 0.757.$$

Si ottiene pertanto la disuguaglianza

$$\frac{7}{15} + \frac{1}{15 \cdot x} + \frac{x}{14} < 0.757,$$

ossia

$$\frac{15 \cdot x^2 - 60.93 \cdot x + 14}{210 \cdot x} < 0.$$

Poiché possiamo assumere che $x > 0$, le soluzioni della disequazione sono costituite da tutti i valori entro l'intervallo determinato dalle eventuali soluzioni reali dell'equazione di secondo grado:

$$x_{1,2} = \frac{60.93 \pm \sqrt{60.93^2 - 4 \cdot 15 \cdot 14}}{2 \cdot 15} \approx \frac{60.93 \pm 53.60}{30}.$$

Dunque la schedulabilità di T_3 è garantita per $0.25 < p_0 < 3.81$.

In conclusione, p_0 deve essere maggiore di $\max\{0.11, 0.12, 0.25\}$, e deve essere minore di $\min\{3.80, 4.49, 3.81\}$. Pertanto, il sistema è certamente schedulabile con $0.25 < p_0 < 3.80$.

(c) *Nelle condizioni del caso (b), determinare con il test di schedulabilità se le scadenze saranno rispettate con un periodo dello scheduler pari a 2.*

Il valore 2 è entro l'intervallo di schedulabilità garantita determinato al punto precedente, dunque il test di schedulabilità dovrà necessariamente dare risposta positiva.

Utilizziamo il test di schedulabilità per un sistema con scheduler periodico.

Schedulabilità del task T_1: consideriamo il sistema di task equivalente costituito da $T_0 = (2, 1/15)$, $T_{0,2} = (8, 1/5)$, $T_{0,3} = (14, 1/5)$, $T_1' = (6, 1/5 + 1/5)$. La funzione di tempo necessario è

$$w_1(t) = \frac{2}{5} + 2 + \left\lceil \frac{t}{2} \right\rceil \frac{1}{15} + \left\lceil \frac{t}{8} \right\rceil \frac{1}{5} + \left\lceil \frac{t}{14} \right\rceil \frac{1}{5}$$

Risolvendo iterativamente l'equazione $w_1(t) = t$ si ottiene:

$$w_1\left(2 + \frac{2}{5}\right) = 2 + \frac{14}{15}, \quad w_1\left(2 + \frac{14}{15}\right) = 2 + \frac{14}{15}.$$

Poiché $2 + 14/15 < 6$, T_1 è schedulabile.

Schedulabilità del task T_2: consideriamo il sistema di task equivalente costituito da $T_0 = (2, 1/15)$, $T_{0,3} = (14, 1/5)$, $T_1' = (6, 1/5 + 1/5)$, $T_2' = (8, 3/5 + 1/5)$. La funzione di tempo necessario è

$$w_2(t) = \frac{4}{5} + 2 + \left\lceil \frac{t}{2} \right\rceil \frac{1}{15} + \left\lceil \frac{t}{14} \right\rceil \frac{1}{5} + \left\lceil \frac{t}{6} \right\rceil \frac{2}{5}.$$

Risolvendo iterativamente l'equazione $w_2(t) = t$ si ottiene:

$$w_2\left(2 + \frac{4}{5}\right) = 3 + \frac{8}{15}, \quad w_2\left(3 + \frac{8}{15}\right) = 3 + \frac{8}{15}.$$

Poiché $3 + 8/15 < 8$, T_2 è schedulabile.

Schedulabilità del task T_3: consideriamo il sistema di task equivalente costituito da $T_0 = (2, 1/15)$, $T_1' = (6, 1/5 + 1/5)$, $T_2' = (8, 3/5 + 1/5)$, $T_3' = (14, 4 + 1/5)$. La funzione di tempo necessario è

$$w_3(t) = \frac{21}{5} + 2 + \left\lceil \frac{t}{2} \right\rceil \frac{1}{15} + \left\lceil \frac{t}{6} \right\rceil \frac{2}{5} + \left\lceil \frac{t}{8} \right\rceil \frac{4}{5}.$$

Risolvendo iterativamente l'equazione $w_3(t) = t$ si ottiene:

$$w_3\left(6 + \frac{1}{5}\right) = 8, \quad w_3\left(8\right) = 8 + \frac{1}{15}, \quad w_3\left(8 + \frac{1}{15}\right) = 8 + \frac{14}{15}, \quad w_3\left(8 + \frac{14}{15}\right) = 8 + \frac{14}{15}.$$

Poiché $8 + 14/15 < 14$, T_3 è schedulabile.

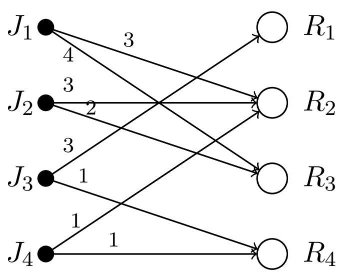
***Esercizio 3**. Un sistema deve eseguire quattro job $J_1, \ldots, J_4$, con J_i di priorità maggiore di J_k se $i < k$. I job utilizzano quattro risorse condivise $R_1, \ldots R_4$, e le relative sezioni critiche sono: $J_1 : [R_2, 3][R_3, 4]$, $J_2 : [R_3, 2][R_2, 3]$, $J_3 : [R_1, 3\,[R_4, 1]\,]$, $J_4 : [R_2, 1][R_4, 1]$.*

***(a)** Quali sono i tempi massimi di blocco per conflitto di risorse utilizzando il protocollo 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:

$$b_1(\mathrm{rc}) = 3; \qquad b_2(\mathrm{rc}) = 3; \qquad b_3(\mathrm{rc}) = 1; \qquad b_4(\mathrm{rc}) = 0.$$

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



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

B_i	J_2	J_3	J_4
J_1			
J_2	$*$		1
J_3		$*$	1

$B_c \equiv B_i$

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:

$$b_1(\mathrm{rc}) = 3; \quad b_2(\mathrm{rc}) = 1; \quad b_3(\mathrm{rc}) = 1; \quad b_4(\mathrm{rc}) = 0.$$

***(c)** Come cambiano i tempi massimi di blocco del precedente punto (b) assumendo che nel sistema sia presente anche un job J_1' di priorità uguale a quella di J_1 e che non utilizza alcuna risorsa condivisa?*

Poiché J_1' non accede ad alcuna risorsa condivisa, non può essere bloccato direttamente dagli altri job e non può influire sui tempi di blocco degli altri job, che restano pertanto invariati. D'altra parte, J_2 e J_4 possono bloccare indirettamente J_1' perché a causa del meccanismo di *inheritance* essi possono assumere temporaneamente la stessa priorità di J_1'. Perciò il massimo tempo di blocco per conflitto di risorse di J_1' è $b_1'(\mathrm{rc}) = 3$.