Sistemi Embedded e Real-time (M. Cesati)

Compito scritto del 12 febbraio 2013

Esercizio 1. Il seguente sistema di task periodici in fase e non interrompibili è schedulato su un processore con un algoritmo "cyclic schedule": $T_1 = (4, 1, 6)$, $T_2 = (6, 1)$, $T_3 = (7, 1)$, $T_4 = (8, 2, 4)$.

(a) Determinare una dimensione appropriata per il frame.

(b) Verificare che non esiste alcuna schedulazione ciclica struttura per l'insieme di task [per la dimensione del frame determinata al punto precedente]

(c) Verificare l'esistenza di una schedulazione ciclica struttura per l'insieme di task ottenuto diminuendo il periodo del task T_3 da 7 a 6, ed allungando la scadenza relativa del task T_4 da 4 a 6.

(d) Assumendo che la schedulazione determinata in (c) si ripeta all'infinito, determinare il tempo di risposta di un job aperiodico rilasciato a $t = 2$ con tempo d'esecuzione $e = 4$ sia nel caso in cui si utilizzi *slack stealing* che nel caso in cui non lo si utilizzi.

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

(a) Supponendo che venga utilizzato uno scheduler EDF 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 EDF sia periodico, che esso controlli la coda dei job pendenti in tempo $e_0 = 3/200$, e che il costo dell'attivazione di un job sia $CS_0 = 1/10$, determinare i possibili valori per il periodo dello scheduler che garantiscono la schedulabilità del sistema.

Esercizio 3. In fase di progettazione di un sistema real-time multiprocessore si prevede di dover eseguire al massimo 2 task periodici $T_i = (2, 1)$, 8 task periodici $T'_i = (3, 1/5)$ e 6 task periodici $T''_i = (5, 1/3)$.

(a) Supponendo di utilizzare uno scheduler partizionato RM-FF, determinare un numero minimo di processori che garantisca il rispetto di tutte le scadenze dei vari task.

(b) Ripetere l'analisi del punto (a) supponendo di utilizzare uno scheduler partizionato EDF-FF.

Sistemi Embedded e Real-time (M. Cesati)

Soluzioni del compito scritto del 12 febbraio 2013

Esercizio 1. *Il seguente sistema di task periodici in fase e non interrompibili è schedulato su un processore con un algoritmo "cyclic schedule":* $T_1 = (4, 1, 6)$, $T_2 = (6, 1)$, $T_3 = (7, 1)$, $T_4 = (8, 2, 4)$.

(a) *Determinare una dimensione appropriata per il frame.*

- Vincolo sui tempi d'esecuzione dei job:

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

- Vincolo sulla divisibilità della lunghezza dell'iperperiodo ($H = \text{mcm}\{4, 6, 7, 8\} = 168$):

$$f \in \{2, 3, 4, 6, 7, 8, 12, 14, 21, 24, 28, 42, 56, 84, 168\}$$

- Vincolo sul task T_1 ($2f - \gcd\{4, f\} \leq 6$):

$$\begin{array}{ll} 2 \cdot 2 - \gcd\{4, 2\} = 2 \leq 6 & \\ 2 \cdot 3 - \gcd\{4, 3\} = 5 \leq 6 & \\ 2 \cdot 4 - \gcd\{4, 4\} = 4 \leq 6 & \\ 2 \cdot 6 - \gcd\{4, 6\} = 10 > 6 & \Rightarrow f \neq 6 \\ 2 \cdot x - \gcd\{4, x\} > 6 \text{ if } x > 6 & \Rightarrow f \leq 6 \end{array}$$

- Vincolo sul task T_2 ($2f - \gcd\{6, f\} \leq 6$):

$$\begin{array}{l} 2 \cdot 2 - \gcd\{6, 2\} = 2 \leq 6 \\ 2 \cdot 3 - \gcd\{6, 3\} = 3 \leq 6 \\ 2 \cdot 4 - \gcd\{6, 4\} = 6 \leq 6 \end{array}$$

- Vincolo sul task T_3 ($2f - \gcd\{7, f\} \leq 7$):

$$\begin{array}{l} 2 \cdot 2 - \gcd\{7, 2\} = 3 \leq 7 \\ 2 \cdot 3 - \gcd\{7, 3\} = 5 \leq 7 \\ 2 \cdot 4 - \gcd\{7, 4\} = 7 \leq 7 \end{array}$$

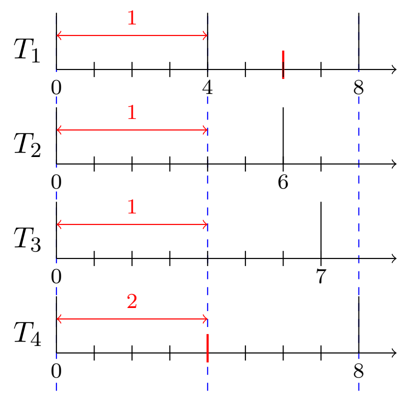
- Vincolo sul task T_4 ($2f - \gcd\{8, f\} \leq 4$):

$$\begin{array}{ll} 2 \cdot 2 - \gcd\{8, 2\} = 2 \leq 4 & \\ 2 \cdot 3 - \gcd\{8, 3\} = 5 > 4 & \Rightarrow f \neq 3 \\ 2 \cdot 4 - \gcd\{8, 4\} = 4 \leq 4 & \end{array}$$

La dimensione per il frame che minimizza l'overhead dello scheduler è $f = 4$.

(b) *Verificare che non esiste alcuna schedulazione ciclica struttura per l'insieme di task [per la dimensione del frame determinata al punto precedente]*

È immediato verificare che non esiste alcuna schedulazione ciclica strutturata per la dimensione del frame $f = 4$. Infatti:



Nel primo ciclo minore dovrebbero essere schedulati tutti i job dei quattro task, con un tempo di esecuzione totale pari a 5, superiore alla quantità di tempo disponibile nel ciclo.

(Per la dimensione $f = 2$ esiste effettivamente una schedulazione ciclica strutturata, che però non è di grande utilità pratica a causa della dimensione del ciclo maggiore.)

(c) *Verificare l'esistenza di una schedulazione ciclica struttura per l'insieme di task ottenuto diminuendo il periodo del task T_3 da 7 a 6, ed allungando la scadenza relativa del task T_4 da 4 a 6.*

La prima conseguenza delle modifiche dei parametri temporali dei task è che la durata dell'iperperiodo (o ciclo maggiore) diminuisce significativamente: $H = \mathrm{mcm}\{4, 6, 8\} = 24$. Le dimensioni del frame ammissibili sono $\{2, 3, 4, 6, 8, 12, 24\}$.

Poiché i parametri temporali del task T_1 non sono cambiati, l'analisi svolta per il punto precedente rimane valida: restano ammissibili solo le dimensioni $\{2, 3, 4\}$. Stessa considerazione può essere svolta per il task T_2.

Per il vincolo sul task T_3 ($2f - \gcd\{6, f\} \leq 7$):

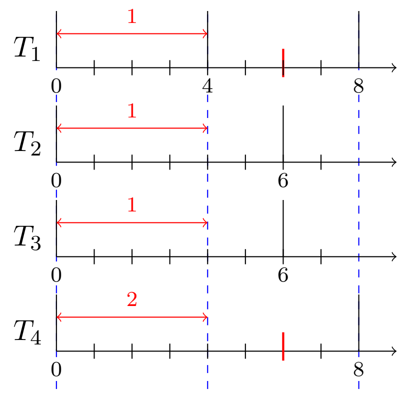
$$\begin{aligned} 2 \cdot 2 - \gcd\{6, 2\} = 2 \leq 6 \\ 2 \cdot 3 - \gcd\{6, 3\} = 3 \leq 6 \\ 2 \cdot 4 - \gcd\{6, 4\} = 6 \leq 6 \end{aligned}$$

Infine per il vincolo sul task T_4 ($2f - \gcd\{8, f\} \leq 6$):

$$
\begin{aligned}
2 \cdot 2 - \gcd\{8, 2\} = 2 \leq 6 \\
2 \cdot 3 - \gcd\{8, 3\} = 5 \leq 6 \\
2 \cdot 4 - \gcd\{8, 4\} = 4 \leq 6
\end{aligned}
$$

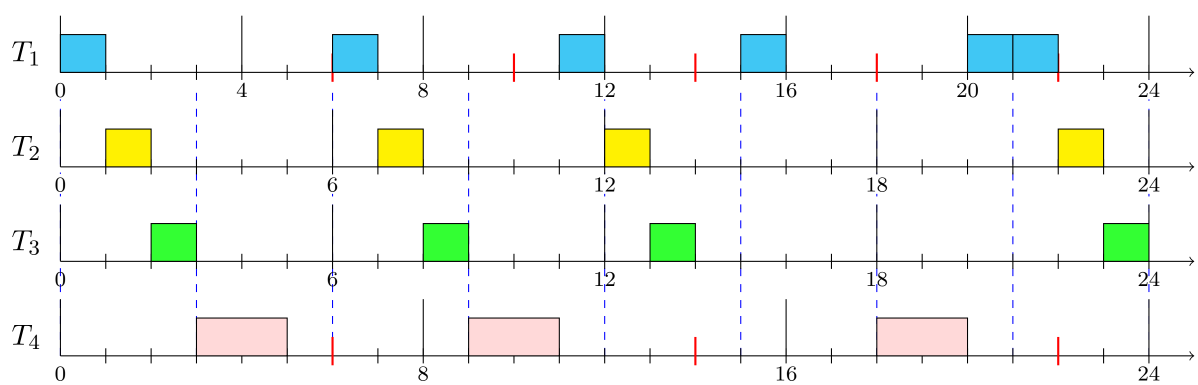
Le dimensioni ammissibili per il frame sono pertanto $\{2, 3, 4\}$.

È immediato verificare che non esiste alcuna schedulazione ciclica strutturata per la dimensione del frame $f = 4$. Infatti:



Nel primo ciclo minore dovrebbero essere schedulati tutti i job dei quattro task, con un tempo di esecuzione totale pari a 5, superiore alla quantità di tempo disponibile nel ciclo.

D'altra parte per $f = 3$ è facile determinare una schedulazione ciclica strutturata:



(d) *Assumendo che la schedulazione determinata in (c) si ripeta all'infinito, determinare il tempo di risposta di un job aperiodico rilasciato a* $t = 2$ *con tempo d'esecuzione* $e = 4$ *sia nel caso in cui si utilizzi* slack stealing *che nel caso in cui non lo si utilizzi.*

Negli otto cicli minori (frame) all'interno del ciclo maggiore vi sono quattro unità di *slack*, rispettivamente nel secondo (1 unità), quinto (1 unità) e sesto ciclo (2 unità). Poiché il job aperiodico è rilasciato a $t = 2$, quindi all'interno del primo frame, lo scheduler assegnerà il processore al job aperiodico nei frame inizianti agli istanti $t = 3, 12, 15$. Il job aperiodico si conclude dunque nel sesto frame iniziante a $t = 15$.

Il tempo di risposta del job aperiodico dipende esclusivamente dalla schedulazione all'interno del sesto frame. In presenza di *slack stealing* lo scheduler anticipa l'esecuzione del job aperiodico rispetto a quella dei task periodici; perciò il job termina a $t = 17$. Invece in assenza di *slack stealing* il job termina alla fine del frame, ossia a $t = 18$. I tempi di risposta sono pertanto:

- Con *slack stealing*: $17 - 2 = 15$ unità di tempo
- Senza *slack stealing*: $18 - 2 = 16$ unità di tempo

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

(a) *Supponendo che venga utilizzato uno scheduler EDF 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

$$\frac{1}{2} + \frac{1}{15} + \frac{1}{6} = \frac{11}{15}.$$

Poiché $11/15 < 1 = U_{\text{EDF}}$, possiamo concludere immediatamente che il sistema di task è schedulabile.

(b) *Supponendo che lo scheduler EDF sia periodico, che esso controlli la coda dei job pendenti in tempo $e_0 = 3/200$, e che il costo dell'attivazione di un job sia $CS_0 = 1/10$, determinare i possibili valori per il periodo dello scheduler che garantiscono 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

$$T_0 = (p_0, 3/200),\ \ T_1' = (2, 11/10),\ \ T_2' = (3, 3/10),\ \ T_3' = (6, 11/10).$$

Il fattore di utilizzazione del sistema di task equivalente è:

$$U_{T'} = \frac{3}{200 \cdot p_0} + \frac{11}{20} + \frac{1}{10} + \frac{11}{60} = \frac{3}{200 \cdot p_0} + \frac{5}{6}.$$

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:

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

È sufficiente garantire la schedulabilità del task T_1'. Infatti, se T_1' fosse schedulabile, allora lo sarebbero certamente anche tutti gli altri task, in quanto hanno lo stesso identico tempo di blocco di T_1' e periodi più grandi.

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

$$\frac{e_1'}{p_1} + \frac{e_2'}{p_2} + \frac{e_3'}{p_3} + \frac{e_0}{p_0} + \frac{p_0}{p_1} \le 1.$$

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

$$\frac{5}{6} + \frac{3}{200 \cdot x} + \frac{x}{2} \le 1,$$

ossia

$$\frac{300 \cdot x^2 - 100 \cdot x + 9}{600 \cdot x} \le 0.$$

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:

$$x_{1,2} = \frac{100 \pm \sqrt{100^2 - 4 \cdot 300 \cdot 9}}{2 \cdot 300} = \frac{100 \pm \sqrt{-800}}{600}.$$

Poiché l'equazione non ammette soluzioni reali, concludiamo che non esiste alcun valore per il periodo p_0 che garantisce la schedulabilità del sistema.

Esercizio 3. *In fase di progettazione di un sistema real-time multiprocessore si prevede di dover eseguire al massimo 2 task periodici $T_i = (2, 1)$, 8 task periodici $T_i' = (3, 1/5)$ e 6 task periodici $T_i'' = (5, 1/3)$.*

(a) *Supponendo di utilizzare uno scheduler partizionato RM-FF, determinare un numero minimo di processori che garantisca il rispetto di tutte le scadenze dei vari task.*

Il fattore di utilizzazione dell'algoritmo di schedulazione partizionato RM-FF è

$$U_{RM\text{-}FF}(m) = m \cdot \left(\sqrt{2} - 1\right)$$

ove m è il numero di processori nel sistema. I task periodici da eseguire hanno un fattore di utilizzazione pari a

$$U_T = 2 \cdot \frac{1}{2} + 8 \cdot \frac{1/5}{3} + 6 \cdot \frac{1/3}{5} = \frac{29}{15}$$

Se $U_T \leq U_{RM\text{-}FF}(m)$, tutte le scadenze dei task sono rispettate, il che significa che m processori sono sufficienti. Si ottiene perciò:

$$m \geq \frac{29/15}{\sqrt{2} - 1} \approx 4.67$$

È quindi necessario includere nel sistema almeno 5 processori.

(b) *Ripetere l'analisi del punto (a) supponendo di utilizzare uno scheduler partizionato EDF-FF.*

Il fattore di utilizzazione dell'algoritmo di schedulazione partizionato EDF-FF è

$$U_{EDF\text{-}FF}(m, \beta) = \frac{\beta \cdot m + 1}{\beta + 1}$$

ove m è il numero di processori e $\beta = \lfloor 1/\max_k\{e_k/p_k\} \rfloor$.

Nel caso in esame, il massimo fattore di utilizzazione dei task è $1/2$, quindi $\beta = 2$. Di conseguenza, un numero sufficiente di processori per garantire la fattibilità del sistema di task si ottiene risolvendo $U_T \leq U_{EDF\text{-}FF}(m, 2)$, ossia:

$$\frac{29}{15} \leq \frac{2\,m + 1}{2 + 1}$$

che equivale a

$$m \geq \frac{3 \cdot 29/15 - 1}{2} = 2 + \frac{2}{5}.$$

È quindi necessario includere nel sistema almeno 3 processori.