DIMOSTRAZIONI

· OTTIMALITÀ RM

Considero due task τ_1 e τ_2 con $T_1 < T_2$.

Se assegnamo le priorità in modo non RM, si avrà $Pri(\tau_1) < Pri(\tau_2)$

La schedule è fattibile se $C_1 + C_2 \leq T_1$, cioè nel periodo del task a più bassa priorità riesco ad eseguire tutto.

Assegno le priorità con RM, cioè $Pri(\tau_1) > Pri(\tau_2)$. Posso avere due situazioni:

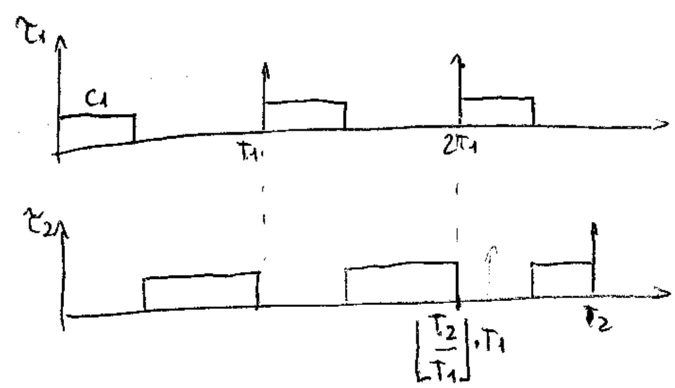
(a) tutte le richieste di τ_1 entro l'intervallo critico di τ_2 sono completate

(b) durante l'ultima richiesta di τ_1 viene rilasciato nuovamente τ_2.

ovvero:

(a) $C_1 \leq T_2 - \left\lfloor \frac{T_2}{T_1} \right\rfloor T_1$

(b) $C_1 \geq T_2 - \left\lfloor \frac{T_2}{T_1} \right\rfloor T_1$



Caso (a)

La schedule è fattibile se

$$\left(\left\lfloor \frac{T_2}{T_1} \right\rfloor + 1\right) C_1 + C_2 \leq T_2$$

che posso ricavare da:

$$C_1 + C_2 \leq T_1 \rightarrow \left\lfloor \frac{T_2}{T_1} \right\rfloor C_1 + \left\lfloor \frac{T_2}{T_1} \right\rfloor C_2 \leq \left\lfloor \frac{T_2}{T_1} \right\rfloor T_1 \xrightarrow{\left\lfloor \frac{T_2}{T_1} \right\rfloor \geq 1} \left\lfloor \frac{T_2}{T_1} \right\rfloor C_1 + C_2 \leq \left\lfloor \frac{T_2}{T_1} \right\rfloor T_1$$

$\rightarrow$ siccome in (a) si ha $T_2 - C_1 \geq \left\lfloor \frac{T_2}{T_1} \right\rfloor T_1 \rightarrow \left\lfloor \frac{T_2}{T_1} \right\rfloor C_1 + C_2 \leq \left\lfloor \frac{T_2}{T_1} \right\rfloor T_1 \leq T_2 - C_1$

$$\rightarrow \left(\left\lfloor \frac{T_2}{T_1} \right\rfloor + 1\right) C_1 + C_2 \leq T_2 \quad \blacksquare$$

caso (b)

La schedule è fattibile se $\left\lfloor \frac{T_2}{T_1} \right\rfloor C_1 + C_2 \leq \left\lfloor \frac{T_2}{T_1} \right\rfloor T_1$ che posso ricavare da:

$$C_1 + C_2 \leq T_1 \rightarrow \left\lfloor \frac{T_2}{T_1} \right\rfloor C_1 + \left\lfloor \frac{T_2}{T_1} \right\rfloor C_2 \leq \left\lfloor \frac{T_2}{T_1} \right\rfloor T_1 \xrightarrow{\left\lfloor \frac{T_2}{T_1} \right\rfloor \geq 1} \left\lfloor \frac{T_2}{T_1} \right\rfloor C_1 + C_2 \leq \left\lfloor \frac{T_2}{T_1} \right\rfloor T_1 \quad \blacksquare$$

Pertanto, anche la schedule RM è fattibile, valendo anche per lei la condizione $C_1 + C_2 \leq T_1$ imposta da schedule non RM. $\blacksquare$

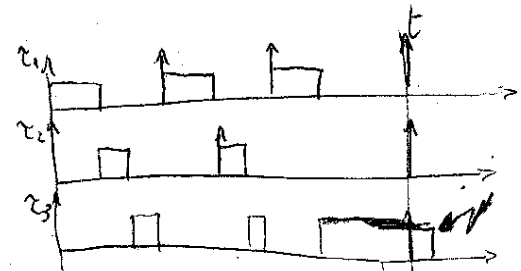
• per task armonici, $U_{LUB}(RM) = 1$.

Supponiamo che τ_i manchi la propria deadline all'istante t, multiplo intero di T_i e di tutti i T_k con $k \le i-1$.
Il tempo totale per completare i Job con deadline t è:

$$\sum_{k=1}^{i} \frac{t}{T_k}\cdot C_k = t\cdot U_i = t\cdot \sum_{k=1}^{i} \frac{C_k}{T_k}$$

se τ_i manca la deadline, vuol dire che

$$t\cdot U_i > t \Rightarrow U_i > 1 \Rightarrow U > 1$$



$$\frac{t}{T_1}\cdot C_1 + \frac{t}{T_2}\cdot C_2 + \frac{t}{T_3}\cdot C_3 = 3C_1 + 2C_2 + C_3$$

• BOUND (condizioni sufficienti) DM, RM

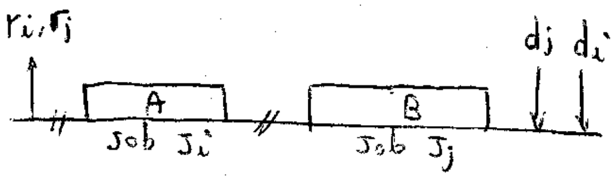
↳ Liu-Layland $\quad U(RM) \le N\cdot\left(2^{1/N} - 1\right)$ con N = n° task

↳ Kuo-Mok $\quad U(RM) \le N_h\cdot\left(2^{1/N_h} - 1\right)$ con N_h = n° sottoinsiemi di task armonici
S': $T' = \min\{T_i\}$
$U' = \sum_i U_i$

↳ Iperbolico $\quad \prod_{(J=1,N)} \left(1 + U_J\right) \le 2$

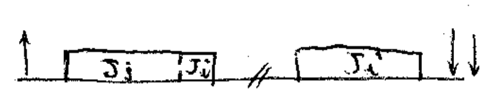
• OTTIMALITÀ EDF (preemption e monoprocessore)

Supponiamo che una schedule S rispetti i vincoli temporali mentre la schedule EDF non li rispetti. Accade che:



Possono succedere due situazioni:

(1) $L(A) > L(B)$. Allora eseguo J_j in una parte di A e J_i nella parte rimanente di A e in B



(2) $L(B) > L(A)$. Allora eseguo J_j in A e in una parte di B e J_i nella parte rimanente di B.

In entrambi i casi, la schedule è schedulata in modo EDF ed è fattibile.
Ripeto il ragionamento per tutte le coppie di Job schedulate in modo non EDF.

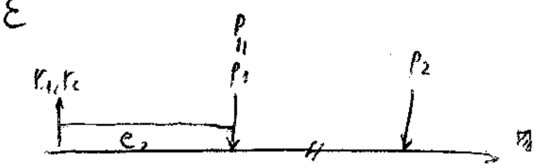
- $U_{LUB}(FIFO) = 0$

Dato un qualunque livello di utilizzazione $\varepsilon > 0$ è possibile trovare un task set con utilizzazione ε non schedulabile in modo FIFO. Ad esempio:

$$T_1 = \left(P, \varepsilon \cdot \frac{P}{2}\right) \qquad T_2 = \left(\frac{2P}{\varepsilon}, P\right)$$

$$U_1 = \frac{\varepsilon \frac{P}{2}}{P} = \frac{\varepsilon}{2} \qquad U_2 = \frac{P}{\frac{2P}{\varepsilon}} = \frac{\varepsilon}{2} \qquad U = U_1 + U_2 = \varepsilon$$

Se T_1 arriva appena dopo T_2, manca la deadline:

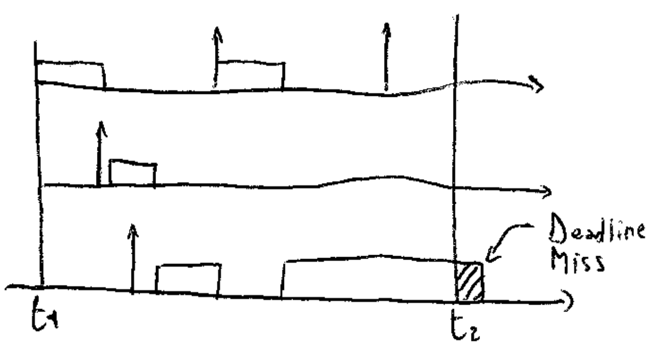


- OTTIMALITÀ DM

Si consideri un insieme di task $T = \{\tau_1, \tau_2, \dots, \tau_n\}$ a priorità decrescente. Per qualche task $D_K > D_{K+1}$. L'insieme di task è schedulabile.
Consideriamo l'intervallo critico di τ_{K+1}: è la situazione peggiore per tutti gli algoritmi a priorità statica.
Scambiamo la priorità di τ_K e τ_{K+1}: l'insieme resta schedulabile. ■

- $U_{LUB}(EDF) = 1$

Supponiamo che un insieme di task non sia schedulabile da EDF pur avendo $U \leq 1$. Si verifica un overflow in t_2 (deadline miss). In t_1 viene rilasciato un Job.



$$C_p(t_1, t_2) = \sum_i \left\lfloor \frac{t_2 - t_1}{T_i} \right\rfloor \cdot C_i \leq \sum_i \frac{t_2 - t_1}{T_i} \cdot C_i = (t_2 - t_1) U$$

Siccome si ha overflow, vuol dire che il tempo a disposizione non è sufficiente:

$$t_2 - t_1 < C_p(t_1, t_2) \leq (t_2 - t_1) U \Rightarrow U > 1 \text{ assurdo!}$$

- BOUND EDF (condizioni sufficienti)

↳ $D = T$: $U(EDF) \leq 1$

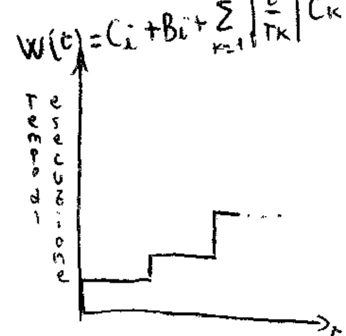
↳ $D \neq T$: $\displaystyle\sum_{i=1}^{n} \frac{C_i}{\min(D_i, T_i)} \leq 1 \xrightarrow{\text{porzioni non revocabili}} \sum_{K=1}^{n} \frac{C_K}{\min(D_K, T_K)} + \frac{B_i}{\min(D_i, T_i)} \leq 1$

- ALGORITMO DI AUDSLEY (calcolo tempo di risposta)

$$\begin{cases} R_i^0 = C_i \\ R_i^s = C_i + \displaystyle\sum_{K=1}^{i-1} \left\lceil \frac{R_i^{s-1}}{T_K} \right\rceil C_K \end{cases}$$

$R_i^s = R_i^{s-1} \rightarrow$ schedulabile

$R_i^s > D_i \rightarrow$ non schedulabile



- SCHEDULAZIONE IN BACKGROUND

Idle time disponibile : $\Phi = (1-U) \cdot H$

Il job aperiodico $J_A (C_A, D_A)$ è schedulabile in background se (sufficiente)

$$\left\lceil \frac{C_A}{\Phi} \right\rceil \cdot H \leq D_A \quad \text{con } D_A \geq H$$

Per m job aperiodici ordinati in modo EDF, l'insieme è garantito al tempo t se:

$$\forall h = 1, \dots, m \quad : \quad t + \left\lceil \frac{s_h}{\Phi} \right\rceil H \leq d_h$$

con $S_h = \sum_{i=1}^{h} c_i(t)$ e $c_i(t)$ tempo residuo di esecuzione del job i all'istante t.

- POLLING SERVER

Il test di garanzia più generale è:

$$\sum_{i \text{ period.}} \frac{C_i}{T_i} + \sum_{j \text{ sporad.}} \frac{C_j}{D_j} + \underbrace{\frac{C_s}{T_s}}_{\text{poller}} \leq U_{LUB}$$

- DEFERRABLE SERVER

Il test di garanzia (sufficiente) è iterativo:

$$\underbrace{\sum_{j=1}^{i} \frac{C_j}{T_j}}_{\text{task a priorità maggiore + } U_i} + \underbrace{\frac{C_s}{T_s}}_{\text{Deferrable Server}} + \underbrace{\frac{C_s}{T_i}}_{\text{Massimo tempo di blocco}} \leq U_{RM}(i+1)$$

oppure test unico:

$$\sum_{i=1}^{N} \frac{C_i}{T_i} + \frac{C_s}{T_s} + \frac{C_s}{T_{k_{min}}} \leq U_{RM}(N+1)$$

$T_{k_{min}} \rightarrow$ periodo del task a max priorità tra quelli con pri(i) < pri(j)