Sistemi Real Time

Ubaldo Vitiello

25 ottobre 2025

Capitolo 1

Esercizi Prima Intercorso

Esercizio RM e priority inheritance con Liu e Layland. Dato l'insieme di task periodici in tabella, in cui, per ogni task si ipotizza l'uso di al massimo 4 sezioni critiche sui semafori S_1, S_2, S_3, ed S_4 ed in cui i valori nelle colonne S_i rappresentano le durate massime delle rispettive sezioni critiche, si determini se l'insieme è schedulabile con RM e priority inheritance, utilizzando il test di Liu e Layland

τ	C_i	T_i	S_1	S_2	S_3	S_4
J_1	15	70	2	4	5	0
J_2	6	20	0	2	0	4
J_3	10	135	1	3	0	5
J_4	5	30	3	0	0	1

Ordiniamo i task in ordine di priorità:

τ	C_i	T_i	S_1	S_2	S_3	S_4
J_2	6	20	0	2	0	4
J_4	5	30	3	0	0	1
J_1	15	70	2	4	5	0
J_3	10	135	1	3	0	5

Calcoliamo i B_i per ogni task. Partiamo da B_2, ricordando che vale:

τ	C_i	T_i	S_1	S_2	S_3	S_4
J_2	6	20	0	2	0	4
J_4	5	30	3	0	0	1
J_1	15	70	2	4	5	0
J_3	10	135	1	3	0	5

Selezioniamo solo i semafori con ceiling maggiore o uguale alla priorità di J_2, cioè i semafori S_2 e S_4:

τ	C_i	T_i	S_1	S_2	S_3	S_4
J_2	6	20	0	2	0	4
J_4	5	30	3	0	0	1
J_1	15	70	2	4	5	0
J_3	10	135	1	3	0	5

Ora selezioniamo di questi semafori solo le righe sotto la riga di J_2:

τ	C_i	T_i	S_1	S_2	S_3	S_4
J_2	6	20	0	2	0	4
J_4	5	30	3	0	0	1
J_1	15	70	2	4	5	0
J_3	10	135	1	3	0	5

τ	C_i	T_i	S_1	S_2	S_3	S_4
J_2	6	20	0	2	0	4
J_4	5	30	3	0	0	1
J_1	15	70	2	4	5	0
J_3	10	135	1	3	0	5

Ora calcoliamo B_4:

$$B_4^l = 4 + 5 = 9$$

$$B_4^s = 2 + 4 + 5 = 11$$

Dunque si ha che B_4 vale:

$$B_4 = \min\left\{B_4^l, B_4^s\right\} = \min\{9, 11\} = 9$$

Ora calcoliamo B_1:

$$B_1^l = 5$$

$$B_1^s = 1 + 3 + 0 + 5 = 9$$

Dunque si ha che B_1 vale:

$$B_1 = \min\left\{B_1^l, B_1^s\right\} = \min\{5, 9\} = 5$$

Infine B_3 vale:

$$B_3 = 0$$

Verifichiamo la schedulabilità con il test di Liu e Layland, che ricordiamo essere una condizione solo sufficente.
Si ha che:

$$\forall i, 1 \leq i \leq n, \sum_{k=1}^{i} \frac{C_k}{T_k} + \frac{B_i}{T_i} \leq i(2^{1/i} - 1)$$

In cui in questa formula i task dovrebbero essere ordinati in ordine di priorità, con 1 il più prioritario ed n il meno prioritario. Andiamo ad applicare la formula per ogni i:

$$J_2 : \frac{C_2}{T_2} + \frac{B_2}{T_2} \leq 1(2^{1/1} - 1) = 1$$

$$J_2 : \frac{6}{20} + \frac{9}{20} = \frac{15}{20} \leq 1$$

$$J_4 : \frac{C_2}{T_2} + \frac{C_4}{T_4} + \frac{B_4}{T_4} \leq 2(2^{1/2} - 1) = 0.828$$

$$J_4 : \frac{6}{20} + \frac{5}{30} + \frac{9}{30} = 0.766 \leq 0.828$$

$$J_1 : \frac{C_2}{T_2} + \frac{C_4}{T_4} + \frac{C_1}{T_1} + \frac{B_1}{T_1} \leq 3(2^{1/3} - 1) = 0.7798$$

$$J_1 : \frac{6}{20} + \frac{5}{30} + \frac{15}{70} + \frac{5}{70} = 0.7523 \leq 0.7798$$

$$J_3 : \frac{C_2}{T_2} + \frac{C_4}{T_4} + \frac{C_1}{T_1} + \frac{C_3}{T_3} + \frac{B_3}{T_3} \leq 4(2^{1/4} - 1) = 0.7568$$

$$J_3 : \frac{6}{20} + \frac{5}{30} + \frac{15}{70} + \frac{10}{135} = 0.7550 \leq 0.7568$$

Dunque l'insieme di task è schedulabile con RM e priority inheritance.

Esercizio RM e priority inheritance con Liu e Layland. Consideriamo i seguenti task periodici:

τ	C_i	T_i	S_1	S_2	S_3	S_4	S_5
J_1	5	28	0	0	2	1	0
J_2	3	16	1	0	0	2	0
J_3	4	60	1	0	2	3	0
J_4	10	40	3	4	0	0	1
J_5	10	100	0	2	1	0	4

Per prima cosa bisogna riscrivere i task in ordine di priorità, in cui un task con periodo minore (quindi con deadline minore) ha priorità maggiore rispetto ad un task con periodo maggiore. Dunque la nuova tabella diventa:

Tabella 1.1: Task ordinati per priorità

τ	C_i	T_i	S_1	S_2	S_3	S_4	S_5
J_2	3	16	1	0	0	2	0
J_1	5	28	0	0	2	1	0
J_4	10	40	3	4	0	0	1
J_3	4	60	1	0	2	3	0
J_5	10	100	0	2	1	0	4

Calcoliamo i B_i. Partiamo da B_2, ricordando che vale:

$$B_2 = \min\left\{B_1^l, B_2^s\right\}$$

Calcoliamo B_1^l. Notiamo per prima cosa che $C(J_2) = P_2$, andiamo quindi a selezionare solo i semafori con $C(S_i) \geq P_2$, cioè S_1 e S_4. Per vederlo graficamente dobbiamo selezionare solo i semafori che hanno almeno un elemento diverso da 0 sopra la linea rossa orizzontale che rappresenta la priorità J_2:

Tabella 1.2: Task ordinati per priorità

τ	C_i	T_i	S_1	S_2	S_3	S_4	S_5
J_2	3	16	1	0	0	2	0
J_1	5	28	0	0	2	1	0
J_4	10	40	3	4	0	0	1
J_3	4	60	1	0	2	3	0
J_5	10	100	0	2	1	0	4

Dunque i semafori che hanno almeno un elemento diverso da 0 sono S_1 e S_4:
Ora andiamo a selezionare di questi semafori solo le righe sono la riga di J_2:
Ora per calcolare B_2^l per ogni riga sotto J_2 andiamo a fare il massimo solo tra gli elementi delle colonne dei semafori selezionati, cioè S_1 e S_4:

τ	C_i	T_i	S_1	S_2	S_3	S_4	S_5
J_2	3	16	1	0	0	2	0
J_1	5	28	0	0	2	1	0
J_4	10	40	3	4	0	0	1
J_3	4	60	1	0	2	3	0
J_5	10	100	0	2	1	0	4

Tabella 1.3: Task ordinati per priorità

τ	C_i	T_i	S_1	S_2	S_3	S_4	S_5
J_2	3	16	1	0	0	2	0
J_1	5	28	0	0	2	1	0
J_4	10	40	3	4	0	0	1
J_3	4	60	1	0	2	3	0
J_5	10	100	0	2	1	0	4

τ	C_i	T_i	S_1	S_2	S_3	S_4	S_5	max
J_2	3	16	1	0	0	2	0	
J_1	5	28	0	0	2	1	0	1
J_4	10	40	3	4	0	0	1	3
J_3	4	60	1	0	2	3	0	3
J_5	10	100	0	2	1	0	4	0

Tabella 1.4: Massimi per calcolo B_i^l

B_2^l è dato dalla somma dei massimi:

$$B_2^l = 1 + 3 + 3 + 0 = 7$$

Ora calcoliamo B_2^s. Per calcolarlo dobbiamo considerare come prima tutti i semafori per cui si ha $C(S_i) \geq P_2$, che come prima sono solo S_1 e S_4. Di questi semafori consideriamo solo le righe che stanno sotto J_2:
Ora per ogni semaforo selezionato (dunque per ogni colonna selezionata) andiamo a fare il massimo tra gli elementi del semaforo (aggiungiamo una riga per fare vedere i massimi):
B_2^s è dato dalla somma dei massimi:

$$B_2^s = 3 + 3 = 6$$

Dunque B_2 sarà:

$$B_2 = \min\left\{B_2^l, B_2^s\right\} = \min\{7, 6\} = 6$$

Esercizio RM e priority inheritance con Liu e Layland. Dato l'insieme di task in tabella, in cui, per ogni task si riporta la richiesta di utilizzo massima di 4 risorse R1, R2, R3, ed R4 disponibili in un massimo di 4, 2, 3 e 1 unità, si calcoli il ceiling C(m) per tutte e tre le risorse, al variare del numero di risorse disponibili. Nel caso in cui siano correttamente disponibili 3 unità di R1, 1 unità di R2, 0 unità di R3 e 1 unità di R4. Quale sarà il valore assunto dal ceiling del sistema

Esercizio RM e priority inheritance con Liu e Layland. Dato l'insieme di task periodici in tabella, in cui, per ogni task si ipotizza l'uso di al massimo 4 sezioni critiche sui semafori S_1, S_2, S_3, ed S_4 ed in cui i valori nelle colonne S_i rappresentano le durate massime delle rispettive sezioni critiche, si determini se l'insieme è schedulabile con EDF e Stack Resource Policy (SRP) con il test di Liu e

τ	C_i	T_i	S_1	S_2	S_3	S_4	S_5
J_2	3	16	1	0	0	2	0
J_1	5	28	0	0	2	1	0
J_4	10	40	3	4	0	0	1
J_3	4	60	1	0	2	3	0
J_5	10	100	0	2	1	0	4

τ	C_i	T_i	S_1	S_2	S_3	S_4	S_5
J_2	3	16	1	0	0	2	0
J_1	5	28	0	0	2	1	0
J_4	10	40	3	4	0	0	1
J_3	4	60	1	0	2	3	0
J_5	10	100	0	2	1	0	4
max			3			3	

	D_i	μ_{R1}	μ_{R2}	μ_{R3}	μ_{R4}
J_1	35	4	1	0	1
J_2	10	3	1	0	0
J_3	140	2	0	3	1
J_4	80	1	2	2	0

Tabella 1.5: Richieste di task e disponibilità di risorse

Layland, assumendo che i task con periodo minore abbiano maggior livello di preemption. I tempi di bloccaggio possono essere calcolati con il metodo che si adotta per il priority ceiling.

G	T_i	C_i	S_1	S_2	S_3	S_4
J_1	5	25	4	0	0	0
J_2	12	150	0	5	0	6
J_3	4	16	0	2	0	0
J_4	2	10	0	0	1	0
J_5	16	60	4	0	3	8

Esercizio RM e priority inheritance con Liu e Layland. Si considerino i task periodici indicati in tabella, con deadline relativa inferiore al periodo. Ciascun task utilizza al massimo di 2 risorse critiche protette rispettivamente dai semafori S_1 ed S_2 per i quali si riporta la durata massima delle sezioni critiche per ogni task. Supponendo che i task siano schedulati con algoritmo EDF e le risorse gestite con SRP, si verifichi la fattibilità dell'insieme dei task adottando il metodo del *processor demand criterion.* Il calcolo dei tempi di bloccaggio può essere fatto con lo stesso metodo che si utilizza per il protocollo di priority ceiling.

τ	C_i	D_i	T_i	S_1	S_2
τ_1	4	6	20	0	0
τ_2	1	3	4	1	0
τ_3	2	8	10	0	2
τ_4	1	7	8	1	1

Esercizio Timeline Scheduling. Si consideri l'insieme di task periodici caratterizzati dalla frequenza di attivazione (in Hz) e dal tempo di calcolo (in ms), riportati nella tabella precedente. Ipotizzando

Tabella 1.6: Dati dei task (frequenza di attivazione e tempo di calcolo)

	T_1	T_2	T_3	T_4	T_5
f (Hz)	25	100	12,5	6,25	50
C_i (ms)	3	2	4	7	6

l'uso del Timeline Scheduling (Cyclic Executive), si calcolino il minor cycle (o time slot) e il major cycle (o iperperiodo) e si riporti graficamente la schedulazione ottenuta allocando opportunamente le istanze dei task nei time slot, ipotizzando che non vi siano vincoli sul jitter di attivazione dei job. Si riportino (e testino) anche le condizioni da verificare per assicurare la schedulabilità dei task nei time slot. Si ipotizzi infine che il tempo di calcolo del task T_5 passi da 6 ms a 9 ms, rendendo la schedulazione non fattibile: quale soluzione adottare in questo caso? Discutere la possibile soluzione mostrando graficamente come si modifica la timeline e riportando le nuove condizioni da verificare.

Esercizio Processor Demand Criterion. Si consideri l'insieme di task periodici rappresentato nella tabella. Si determini se l'insieme è schedulabile con EDF mediante il test di Liu e Layland. Si ripeta poi il test attraverso il processor demand criterion, rappresentando graficamente la domanda di calcolo nell'iperperiodo al variare del tempo L, e verificando che essa sia sempre inferiore alla funzione identità $g(L) = L$.

Tabella 1.7: Dati dei task

	τ_1	τ_2	τ_3
C_i	2	3	6
D_i	5	9	13
T_i	6	12	18

$$\sum_{i=1}^{n} \frac{C_i}{D_i} \leq n(2^{1/n} - 1)$$

$$\frac{C_1}{D_1} + \frac{C_2}{D_2} + \frac{C_3}{D_3} \leq 3(2^{1/3} - 1)$$

$$\frac{2}{5} + \frac{3}{9} + \frac{6}{13} = 1.194 \geq 0.7798$$

Dunque il test di Liu e Layland non è superato, ma esso è solo un test sufficiente, quindi procediamo con il processor demand criterion.

Calcoliamo U:

$$\sum_{i=1}^{n} \frac{C_i}{T_i} = \frac{2}{6} + \frac{3}{12} + \frac{6}{18} = 0.9166$$

Calcoliamo L^*:

$$L^* = \frac{1}{1-U} \sum_{i=1}^{n} (T_i - D_i) U_i$$

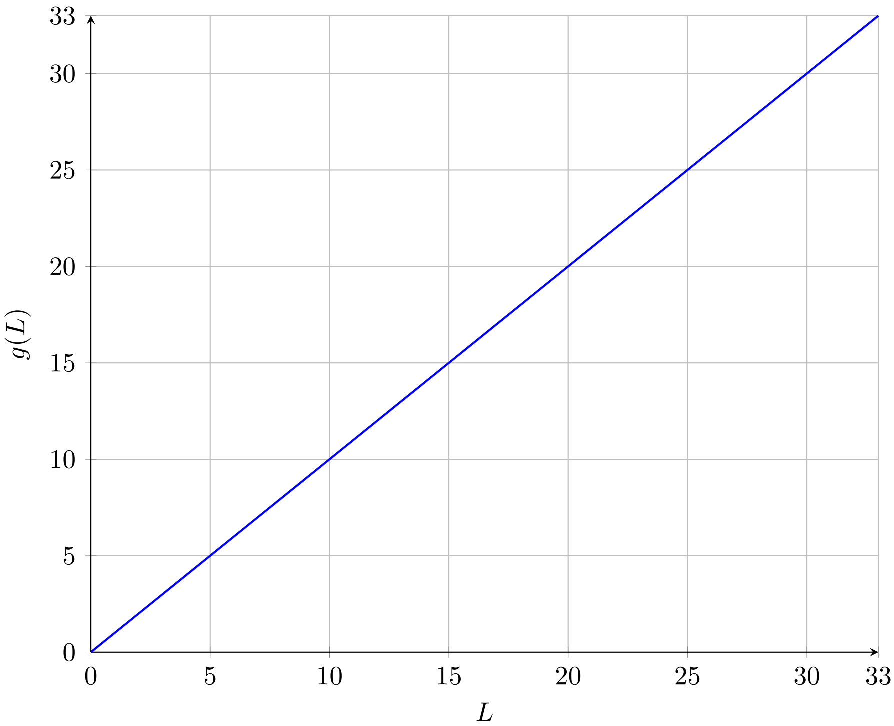
In cui $U_i = \dfrac{C_i}{T_i}$, dunque:

$$L^* = \frac{1}{1 - 0.9166} \left((6-5)\frac{2}{6} + (12-9)\frac{3}{12} + (18-13)\frac{6}{18} \right) = 32.97$$

Calcoliamo l'iperperiodo:

$$H = m.c.m.(6, 12, 18) = 36$$

z¡



Esercizio Deadline Monotonic con Liu e Layland e Response Time Analysis. Si consideri l'insieme di task periodici in tabella, con deadline D_i minore del periodo T_i. Si determini se l'insieme è schedulabile con Deadline Monotonic (DM) attraverso il test di Liu e Layland. Si ripeta poi il test attraverso il metodo dell'interferenza I_i e si discutano le differenze rispetto al test precedente, ove presenti.

Tabella 1.8: Dati dei task

	τ_1	τ_2	τ_3	τ_4	τ_5
C_i	2	3	1	5	2
D_i	11	15	8	37	10
T_i	11	18	9	40	12

Eseguiamo per prima cosa il test di Liu e Layland per DM:

$$\sum_{i=1}^{n} \frac{C_i}{D_i} \leq n(2^{1/n} - 1)$$

$$\frac{2}{11} + \frac{3}{15} + \frac{1}{8} + \frac{5}{37} + \frac{2}{10} = 0.841 \geq 5(2^{1/5} - 1) = 0.7435$$

L'ordine di priorità dei task è:

$$\tau_3 > \tau_5 > \tau_1 > \tau_2 > \tau_4$$

Calcoliamo l'interferenza per ogni task:

$$I_3 = 0$$

$$I_5 = \lceil \frac{D_5}{T_3} \rceil C_3 = \lceil \frac{10}{9} \rceil \cdot 1 = 2$$

$$I_1 = \lceil \frac{D_1}{T_5} \rceil C_5 + \lceil \frac{D_1}{T_3} \rceil C_3 = \lceil$$

$$I_2 = \lceil \frac{D_2}{T_1} \rceil C_1 + \lceil \frac{D_2}{T_5} \rceil C_5 + \lceil \frac{D_2}{T_3} \rceil C_3 =$$

$$I_4 = \lceil \frac{D_4}{T_2} \rceil C_2 + \lceil \frac{D_4}{T_1} \rceil C_1 + \lceil \frac{D_4}{T_5} \rceil C_5 + \lceil \frac{D_4}{T_3} \rceil C_3 =$$

Esercizio Deadline Monotonic con Liu e Layland e test d'Interferenza.