

Sistemi Real Time

Ubaldo Vitiello

22 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	T_i	C_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

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:

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

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 :

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

Ora andiamo a selezionare di questi semafori solo le righe sono la riga di J_2 :

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

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	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 :

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

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):

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	

B_2^s è dato dalla somma dei massimi:

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

Dunque B_2 sarà:

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

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

	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

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

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

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

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