

- 1) Discutere i vantaggi e gli svantaggi degli algoritmi di scheduling priority-driven statici e dinamici (max 1 pagina)
- 2) Dato l'insieme di task periodici schedulati in modo RM: $TS=\{\tau_1=(5,0.75), \tau_2=(8,2), \tau_3=(10,1.5), \tau_4=(18,4)\}$ (tutti i parametri sono espressi in ms), determinare argomentando appropriatamente la risposta, se l'insieme di task è schedulabile (si, no, forse), specificando in tutti i casi quali task sono individualmente garantiti (si, no) in base ai seguenti criteri:
 - a. Bound di Liu e Layland
 - b. Bound di Kuo e Mok

Si supponga poi che il task τ_1 sia costituito da una sezione di codice interamente non revocabile e che il task τ_4 presenti una sezione di codice non revocabile di durata pari a 2.5 ms. In tali ipotesi, quali task risultano essere garantiti in base ai due diversi bound considerati?

- 3) Dimostrare che per un insieme di task periodici $U_{lub}(EDF)=1$.
- 4) Si considerino i seguenti insiemi di task periodici indipendenti: $TS1=\{\tau_1=(10,2,5), \tau_2=(8,3,6)\}$, $TS2=\{\tau_3=(10,2,3), \tau_4=(?,3,6)\}$ (i parametri dei task sono espressi nella forma (T_i, C_i, D_i)). Effettuare i test di garanzia basati su bound di schedulabilità associati agli algoritmi DM ed EDF che è possibile applicare agli insiemi di task $TS1$ e $TS2$, precisando se si tratta di condizioni sufficienti o necessarie e sufficienti. Sempre con tali bound, analizzare separatamente $TS1$ e $TS2$ e determinare per ciascun algoritmo se i due insiemi sono garantiti (si, no), se sono schedulabili (si, no, forse) e, all'interno dell'insieme, se ciascun task è garantito.
- 5) Si consideri un insieme di 5 task periodici interagenti $\tau_1, \tau_2, \tau_3, \tau_4, \tau_5$ in esecuzione su un sistema che supporta le preemption e il protocollo di priority inheritance (PIP). Si supponga che tra i task siano presenti le sezioni critiche $S1, S2, S3$ e $S4$ protette da semafori mutex distinti e prive di annidamenti. Le priorità dei task sono di tipo statico ed attribuite in modo decrescente, con τ_1 a massima priorità.

Gli accessi alle sezioni critiche da parte dei task sono i seguenti:

- τ_1 accede a $S1$ con durata 11ms e a $S3$ con durata 7ms
- τ_2 accede a $S2$ con durata 13ms e a $S1$ con durata 2ms
- τ_3 accede a $S3$ con durata 4ms e poi di nuovo a $S3$ con durata 3ms
- τ_4 accede a $S3$ con durata 6ms, a $S2$ con durata 19ms, a $S1$ con durata 5ms e a $S4$ con durata 6ms
- τ_5 accede a $S2$ con durata 1ms e a $S4$ con durata 2ms

Determinare il numero massimo di situazioni di inversione di priorità N_i e il tempo totale di blocco di caso peggiore B_i che ciascun task può subire (e quindi da considerare per valutare la schedulabilità dell'insieme di task). Riportare le risposte nella tabella sottostante.

N1	B1	N2	B2	N3	B3	N4	B4	N5	B5

- 6) In un sistema di elaborazione in tempo reale sono presenti i seguenti task periodici: $T=\{\tau_1=(5,1), \tau_2=(8,2), \tau_3=(10,2), \tau_4=(16,3), \tau_5=(80,1)\}$ (parametri espressi in ms), schedulati in modo priority-driven statico. Il sistema è inoltre predisposto per la schedulazione in background di job aperiodici con modulo di accettazione. Determinare se il sistema, con un test di costo costante, può accettare e garantire il job aperiodico hard real-time $J_a=(D_a=206, C_a=26)$.