

Esercizio 1 (10 punti)

Un sistema operativo adotta la politica di scheduling dei thread a coda multipla e con prelazione tra le code. Sono presenti tre code tutte con scheduling round-robin, la coda ad alta priorità (coda H) usa un quanto $q=1\text{ms}$, la coda a priorità media (coda M) usa un quanto $q=2\text{ms}$ e la coda a bassa priorità (coda L) usa un quanto $q=3\text{ms}$. Inoltre quando un thread ha un CPU burst inferiore al quanto viene promosso nella coda a più alta priorità e se invece il CPU burst è superiore al quanto viene portato nella coda a più bassa priorità. Quando il thread viene prelazionato viene messo in coda.

Il sistema deve schedare i seguenti thread con tempi di arrivo, coda e uso CPU/IO indicati:

T_1 $T_{\text{arrivo}}=3$ coda M CPU(1ms)/IO(4ms)/CPU(1ms)/IO(4ms)/...

T_2 $T_{\text{arrivo}}=1$ coda M CPU(1ms)/IO(4ms)/CPU(1ms)/IO(4ms)/...

T_3 $T_{\text{arrivo}}=0$ coda M CPU(6ms)/IO(4ms)/CPU(3ms)

T_4 $T_{\text{arrivo}}=2$ coda M CPU(3ms)/IO(4ms)/CPU(3ms)

I thread T_1 e T_2 ripetono l'uso di CPU/IO e terminano (alla fine di un CPU burst) quando T_3 e T_4 sono terminati.

Si determini: il **diagramma di Gantt**, il **tempo di attesa medio**, il **tempo di ritorno medio**, il **tempo di risposta medio** e il numero di **cambi di contesto**. Nel diagramma di gantt indicare per ogni thread in quale coda avviene l'esecuzione del thread.

Esercizio 2 (20 punti)

Si vuole realizzare in java il seguente sistema:

Sono presenti N thread *Generator* che acquisiscono un numero progressivo p da un contatore condiviso, costruiscono un messaggio con il valore p , id del generator ed un valore v e lo inseriscono in una coda limitata di K posizioni ed aspettano un tempo TG prima di generare un nuovo valore.

Sono presenti M thread *Worker* dove ognuno iterativamente preleva il messaggio con progressivo minore dalla coda quindi usa una risorsa A per un tempo TA , quindi acquisisce una risorsa B per un tempo TB e quindi rilascia insieme la risorsa A e B quindi inserisce il messaggio con il risultato ed il proprio ID in una coda finale illimitata. Sono presenti inizialmente NA risorse A e NB risorse B .

È presente un thread *Collector* che preleva il messaggio con progressivo minore dalla coda finale e stampa: il progressivo, l'id del generatore, id del worker che ha elaborato ed il valore iniziale ed il valore finale.

Per facilitare il debugging ogni generatore genera valori progressivi ed il worker prende questo valore e lo moltiplica per il suo id e lo mette come risultato.

Il programma principale fa partire i thread quindi dopo 10 secondi vengono fermati tutti i thread, si deve fare in modo che tutte le risorse acquisite vengano rilasciate. Alla fine il programma principale deve stampare per ogni generatore il numero di valori generati, per ogni worker il numero di messaggi processati ed infine il numero di risorse A e B disponibili.

Realizzare il sistema usando SOLO i **semafori** per la sincronizzazione dei thread.