

Esercizio 1 (10 punti)

Un sistema operativo adotta la politica di scheduling dei thread a code multiple con *prelazione* tra le code. Sono presenti due code una ad alta priorità con scheduling round robin con quanto $q=2$ ms ed una a bassa priorità con scheduling Shortest Remaining Time First.

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

T_1	$T_{\text{arrivo}}=3$ pri=H	CPU(1ms)/IO(5ms)/CPU(1ms)
T_2	$T_{\text{arrivo}}=2$ pri=H	CPU(3ms)/IO(7ms)/CPU(1ms)
T_3	$T_{\text{arrivo}}=0$ pri=L	CPU(4ms)/IO(2ms)/CPU(2ms)
T_4	$T_{\text{arrivo}}=1$ pri=L	CPU(2ms)/IO(1ms)/CPU(1ms)

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. Inoltre indicare in ogni istante il numero di processi in attesa.

Esercizio 2 (20 punti)

Si vuole implementare il seguente sistema:

Sono presenti N thread *Generator* dove ognuno genera una sequenza di numeri interi G_p $p=0,1,2,\dots$ il tempo tra la generazione di un numero ed il successivo è casuale tra 100 ms e 1000 ms.

Sono presenti M thread *Processor*, ogni Processor acquisisce N valori “sincroni” generati dai Generator identificati dallo stesso “ p ” li somma ed inserisce il risultato in una coda di uscita insieme al valore p . Il *Processor* impiega un tempo variabile compreso tra 100 e 1000 ms per elaborare i dati. Il thread *Processor* dovrà aspettare se tutti gli N dati non sono stati generati dai Generator mentre il thread *Generator* dovrà aspettare se il valore precedente non è ancora stato acquisito da un Processor.

Un thread *Extactor* estrae i dati dalla coda di uscita e li stampa, l'estrazione deve essere in ordine progressivo di “ p ”.

Il programma principale deve far partire i thread e dopo 30 secondi deve terminare tutti i thread.

Per semplificare il testing il generatore i -esimo genera i numeri $i*100+p$ $i=0..N-1$

Realizzare in java il sistema descritto usando i **metodi sincronizzati** per la sincronizzazione tra thread.

Per $N=5$ riportare le prime 5 righe di output del programma.