Sistemi Operativi

Prima prova in itinere 26-04-2017 - **Compito A**

Esercizio 1 (12 punti)

Un sistema operativo adotta la politica di scheduling dei thread a code multiple con *prelazione* tra le code. Sono presenti tre code, una a priorità 1 (alta priorità) con scheduling round-robin con quanto q=1ms, una coda a priorità 2 (media priorità) con scheduling round-robin con quanto q=2ms e una coda a priorità 3 (bassa priorità) con scheduling SJF (senza prelazione all'interno della coda).

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

T_1 T_{arrivo}=3 pri=1 CPU(1ms)/IO(1ms)/CPU(2ms)
T_2 T_{arrivo}=2 pri=2 CPU(2ms)/IO(2ms)/CPU(3ms)
T_3 T_{arrivo}=0 pri=3 CPU(4ms)/IO(2ms)/CPU(3ms)
T_4 T_{arrivo}=1 pri=3 CPU(2ms)/IO(1ms)/CPU(1ms)

Si determini:

1. il **diagramma di Gantt**,
2. il **tempo di attesa** medio,
3. il **tempo di ritorno** medio,
4. il **tempo di risposta** medio,
5. il numero di cambi di contesto e
6. gli istanti in cui si ha prelazione tra le code.

Esercizio 2 (18 punti)

In un sistema sono presenti N thread *MakeArrayThread* che producono ognuno un array di M numeri float generati in modo randomico con valore in [0,10). Un thread *SumArrayThread* prende gli N array generati e ne calcola l'array somma e lo stampa. Inoltre gli N thread *MakeArrayThread* ed il thread *SumArrayThread* sono fatti partire in modo periodico ogni 5 secondi. Implementare in Java quanto descritto usando il metodo join per sincronizzare i thread e chiedendo N e M in input da tastiera.