Sistemi Operativi

Compito 21-07-2020

Esercizio 1 (10 punti)

Un sistema operativo adotta la politica di scheduling dei thread a coda multipla con due code, le singole code hanno scheduling con priorità e prelazione all'interno della coda. Lo scheduling tra le code è round robin con quanto di 5ms. Se una coda è vuota il quanto non viene usato e passa all'altra coda. Se un thread viene interrotto al termine del quanto viene rimesso in testa alla coda.

Il sistema deve schedulare i seguenti thread arrivati ai tempi indicati con le priorità indicate e con uso CPU/IO indicati:

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

Considerare numeri di priorità bassa indicativi di alta priorità.

Si determini: il **diagramma di Gantt**, il **tempo di attesa** medio, il **tempo di ritorno** medio, il **tempo di risposta** medio, il numero di cambi di contesto.

Esercizio 2 (20 punti)

Sono presenti N persone e M stanze e in ogni stanza ci può entrare una persona per volta. Ogni persona deve iterativamente visitare K stanze diverse. La stanza da visitare deve essere scelta tra quelle visitate di meno (anche non libere in quel momento). In caso di ricerca concorrente della stanza meno visitata si deve fare in modo che le persone scelgano se possibile stanze diverse, si deve evitare ad esempio che all'inizio tutti tentino di accedere alla stanza 1. In ogni stanza la persona rimane per 100ms.

Dopo 30 secondi terminare tutti i thread e per ogni persona stampare il numero stanze visitate e per ogni stanza il numero di visite ricevute.

Realizzare in Java il sistema descritto usando i semafori per la sincronizzazione tra thread.