Sistemi Operativi

Compito 06-07-2020

Esercizio 1 (10 punti)

Un sistema operativo adotta la politica di scheduling dei thread con priorità e prelazione.

Inoltre ogni 6ms viene ricalcolata la priorità dei thread che viene temporaneamente aumentata di 1 se un thread è stato in attesa per tutti i 6ms e che invece ritorna alla priorità originaria se ha ricevuto tempo di CPU. Quando un thread viene prelazionato 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 pri=1 T_{arr}=2 CPU(2ms)/IO(4ms)/CPU(2ms)
T_2 pri=2 T_{arr}=0 CPU(3ms)/IO(4ms)/CPU(2ms)
T_3 pri=2 T_{arr}=3 CPU(4ms)/IO(4ms)/CPU(4ms)
T_4 pri=3 T_{arr}=0 CPU(1ms)/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 che si muovono nel piano cartesiano. Ogni persona cambia posizione dx,dy con dx e dy numeri casuali in [-10,10). Prima di cambiare posizione la persona deve verificare se nella nuova posizione ha una distanza superiore a 1 da tutti gli altri se questo non accade deve aspettare che questo accada. La posizione iniziale è per tutti la coordinata 0,0. Inoltre aspettare 100ms tra un aggiornamento ed il successivo.

Dopo 30 secondi terminare tutti i thread e per ogni persona stampare: il numero di cambi di posizione effettuati, il numero di volte che è entrato in attesa perché la destinazione era troppo vicino ad una persona, lo spazio totale percorso.

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