Sistemi Operativi

Compito 06-07-2021

Esercizio 1 (10 punti)

Un sistema operativo adotta la politica di scheduling dei thread a code multiple con *prelazione* tra le code. Sono presenti tre code ad alta (H), media (M) e bassa (L) priorità tutte con scheduling round-robin con quanto q=2ms.

Quando un thread ha un CPU burst inferiore alla durata del quanto viene promosso nella coda a priorità superiore a quella attuale (se presente) mentre quando il CPU burst sicuramente eccede la durata del quanto viene subito portato nella coda a priorità inferiore rispetto a quella attuale (se presente). Inoltre quando un thread viene prelazionato da una coda a priorità più elevata questo viene messo in fondo alla coda dei thread pronti della sua coda.

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

T_1 T_{arrivo}=3 pri=M CPU(1ms)/IO(6ms)/CPU(3ms)
T_2 T_{arrivo}=2 pri=M CPU(3ms)/IO(2ms)/CPU(3ms)
T_3 T_{arrivo}=0 pri=M CPU(1ms)/IO(6ms)/CPU(3ms)
T_4 T_{arrivo}=1 pri=M CPU(5ms)/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. Nel diagramma di Gantt indicare anche la priorità alla quale si ha l'uso di CPU del thread.

Esercizio 2 (20 punti)

Si vuole simulare il seguente sistema:

Sono presenti ***N*** Persone (N>100) che devono vaccinarsi presso un centro vaccinale in cui possono stare al massimo ***M*** Persone (M<100). Entrate nel centro vaccinale ci sono **A** addetti che controllano l'accesso ed impiegano *T1* secondi ognuno, nel 5% dei casi la persona ha sbagliato giorno o orario ed esce subito. Successivamente entrano in una coda fino al momento della vaccinazione da parte di ***K*** addetti che impiegano *T2* secondi, fatta la vaccinazione le persone aspettano altri *T3* secondi prima di uscire dal centro vaccinale.

Durante le operazioni mostrare ogni secondo il numero di persone nel centro vaccinale, il numero totale di vaccini fatti dagli operatori e il numero di persone tornate a casa.

Quando tutte le persone hanno terminato calcolare il tempo di attesa medio e tempo massimo speso in più rispetto al tempo minimo di permanenza.

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