Sistemi Operativi

Compito 07-06-2024

Esercizio 1 (10 punti)

Un sistema operativo adotta la politica di scheduling dei thread a coda multipla e con prelazione tra le code. Sono presenti tre code, una ad alta priorità (H) con scheduling round-robin e quanto q=2ms, una a media priorità (M) con scheduling round-robin e quanto q=3ms e una a bassa priorità (L) con scheduling SRTF.

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

T_1 T_{arrivo}=3 coda H CPU(2ms)/IO(6ms)/CPU(2ms)/IO(9ms)/CPU(1ms)

T_2 T_{arrivo}=1 coda M CPU(3ms)/IO(6ms)/CPU(2ms)/IO(6ms)/CPU(2ms)

T_3 T_{arrivo}=2 coda L CPU(3ms)/IO(5ms)/CPU(7ms)

T_4 T_{arrivo}=0 coda L CPU(4ms)/IO(4ms)/CPU(2ms)

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**.

Esercizio 2 (20 punti)

Si vuole realizzare in java il seguente sistema:

Sono presenti N thread *Generator* che iterativamente producono un valore ed attendono X ms, il valore viene inserito in una coda limitata di lunghezza L, associata al generatore.

Sono presenti M thread *ProcessorThread* dove ognuno deve estrarre atomicamente N valori in testa alle N code (aspetta se almeno una coda è vuota ed in caso di richiesta concorrente solo uno può estrarre i valori in testa) inoltre al vettore di N valori deve essere associato un numero progressivo che identifica l'estrazione. Il Processor una volta acquisiti gli N valori lì processa in un tempo variabile in [T,T+D) e inserisce in una coda illimitata il risultato con associato il numero della estrazione.

Infine un thread *PrinterThread* iterativamente preleva da questa coda chiedendo di estrarre i risultati sulla base del numero progressivo (deve aspettare se il risultato ancora non è arrivato) quindi deve stampare il progressivo, gli N valori ed il risultato.

Per facilitare il debugging i *GeneratorThread* generano numeri in sequenza partendo da id+1 (id=0..N-1) *e i ProcessorThread* producono come risultato la somma degli N valori.

Il programma principale fa partire i thread quindi dopo 10 secondi vengono fermati tutti i thread. Alla fine il programma principale deve stampare per ogni *GeneratorThread* il numero di messaggi prodotti ed il totale dei messaggi generati, per ogni ProcessorThread il numero di richieste completate, il numero di messaggi rimasti nelle code ed il numero di array stampati da PrinterThread.

Realizzare il sistema usando i **metodi sincronizzati** per la sincronizzazione dei thread.