Sistemi Operativi

Compito 22-06-2022

Esercizio 1 (10 punti)

Un sistema operativo adotta la politica di scheduling dei thread a code multiple *con prelazione* tra le code. Sono presenti due code una a bassa priorità ed una ad alta priorità entrambe con scheduling round robin con quanto q=3ms. Inoltre se un thread a bassa priorità ha un CPU burst inferiore al quanto viene portato nella coda ad alta priorità e un thread ad alta priorità viene portato a bassa priorità se usa tutto il quanto.

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

T_1 T_{arrivo}=3 pri=L CPU(2ms)/IO(2ms)/CPU(2ms)
T_2 T_{arrivo}=2 pri=L CPU(3ms)/IO(3ms)/CPU(3ms)
T_3 T_{arrivo}=1 pri=L CPU(2ms)/IO(5ms)/CPU(2ms)
T_4 T_{arrivo}=0 pri=L CPU(2ms)/IO(6ms)/CPU(4ms)

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. Inoltre indicare nel digramma di Gantt per ogni uso di CPU la priorità a cui avviene.

Esercizio 2 (20 punti)

Si vuole simulare il seguente sistema:

Sono presenti N produttori di farina che consegnano una quantità F ogni TF secondi in un magazzino che ha capacità massima H (se il magazzino non riesce a contenere la farina da consegnare il produttore aspetta).

Il magazzino contiene anche un serbatoio di acqua che viene rifornito con K litri al secondo ed ha capacità massima A.

Ci sono M produttori di pane che per produrre il pane prelevano una quantità X di farina e Y di acqua e se non sono presenti entrambi aspettano, quindi producono una quantità X*10+Y di pane in un tempo TP e il pane prodotto viene immagazzinato con una capacità massima P. Inizialmente è presente un camion che preleva una quantità fissa Q di pane e consegna ai clienti in un tempo TC.

Il programma principale deve far partire i thread necessari e ogni secondo stampare i livelli dei magazzini e serbatoio, il numero di camion attivi e la quantità totale di pane consegnato. Inoltre se il magazzino del pane supera l'80% della capacità viene aggiunto un camion per le consegne e se il magazzino del pane è vuoto e c'è più di un camion un camion viene fermato.

Terminare tutti i thread dopo 2 minuti.

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

Sistemi Operativi

Compito 22-06-2022

Esercizio 1 (10 punti)

An operating system adopts the multi-queue thread scheduling policy with preemption between queues. There are two queues, one at low priority and one at high priority, both with round robin scheduling with quantum q = 3ms. Moreover if a low priority thread has a CPU burst lower than quantum it is brought into the high priority queue and a high priority thread is brought to low priority if it uses all the quantum.

The system must schedule the following threads with indicated arrival times, priorities and CPU / IO usage:

T_1 T_{arrivo}=3 pri=L CPU(2ms)/IO(2ms)/CPU(2ms)
T_2 T_{arrivo}=2 pri=L CPU(3ms)/IO(3ms)/CPU(3ms)
T_3 T_{arrivo}=1 pri=L CPU(2ms)/IO(5ms)/CPU(2ms)
T_4 T_{arrivo}=0 pri=L CPU(2ms)/IO(6ms)/CPU(4ms)

Determine: the Gantt chart, the average wait time, the average turnaround time, the average response time and the number of context changes. Also indicate in the Gantt chart for each use of CPU the priority to which it occurs.

Esercizio 2 (20 punti)

We want to simulate the following system:

There are N flour producers who deliver a quantity F every TF seconds to a warehouse that has maximum capacity H (if the warehouse cannot contain the flour to be delivered, the producer waits).

The warehouse also contains a water tank which is refilled with K liters per second and has a maximum capacity of A.

There are M bread producers who take X quantity of wheat and Y of water to produce bread and if they are not present both wait, then produce X * 10 + Y quantity of bread in a time TP and the bread produced is stored with a maximum capacity P. Initially there is a truck that picks up a fixed quantity Q of bread and delivers to customers in a TC time.

The main program has to start the necessary threads and every second print the levels of the warehouses, the number of active trucks and the total quantity of bread delivered. Furthermore, if the bread warehouse exceeds 80% of capacity, a delivery truck is added and if the bread warehouse is empty and there is more than one truck, a truck is stopped.

End all threads after 2 minutes.

Implement the described system in java using ***synchronized methods*** for synchronization between threads.