Sistemi Operativi

Compito 25-11-2020

Esercizio 1 (10 punti)

Un sistema operativo adotta la politica di scheduling dei thread basata su uso di CPU, i thread pronti sono in unica coda e ordinati in base al tempo di CPU usato dall'inizio dell'esecuzione e al tempo di permanenza in coda, quindi lo scheduler al momento di decidere il thread da eseguire selezionerà quello che ha usato meno CPU e a parità quello da più tempo in attesa e a parità quello con id minore. Lo scheduling è con prelazione e ad ogni thread viene dato un quanto di 2ms.

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

T_1 T_{arr}=1 CPU(2ms)/IO(4ms)/CPU(3ms)
T_2 T_{arr}=0 CPU(2ms)/IO(4ms)/CPU(2ms)
T_3 T_{arr}=2 CPU(4ms)/IO(2ms)/CPU(2ms)
T_4 T_{arr}=3 CPU(3ms)/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, il **numero di cambi di contesto**.

Esercizio 2 (20 punti)

Un sistema deve gestire delle immagini di disco dove ognuna è fatta da N Layer. Ogni Layer è rappresentato da un file zip che deve essere scaricato da un repository e devono essere estratti (unzip) in sequenza. Sono presenti K *DownloadThread* che iterativamente acquisiscono da una coda i Layer da scaricare ed impiegano un tempo proporzionale alla loro dimensione. I Layer scaricati devono essere messi in una seconda coda dalla quale un *ExtractorThread* preleva i Layer e li estrae facendo in modo di estrarre tutti i Layer in sequenza. Il thread deve attendere se il layer da estrarre ancora non è stato scaricato.

Il programma principale deve creare e far partire i thread quindi generare gli N layer ognuno di dimensioni casuali (1-100) e quindi deve attendere che tutti i layer vengano acquisiti ed estratti e quindi termini tutti i thread.

I *DownloadThread* e l'ExtractorThead devono simulare le operazioni con uno sleep proporzionale alla dimensione del file, size*100ms il download e size*50ms l'estrazione e al termine stampare il numero del layer e l'operazione completata.

Realizzare in Java il sistema descritto usando i semafori per la sincronizzazione tra thread (non usare attesa attiva).