

Tutoraggio di Sistemi Operativi

Matteo Federico
Lezione #6





PowerNap dei thread

Creare un programma che: Dato un numero N , crea N thread, ogni thread dovrà dormire su tutte le cpu del processore.

Su ogni CPU ogni thread dovrà dormire 5ms. quando un thread si sveglia dovrà estrearre randomicamente la prossima cpu su cui dormire! I thread termina quando ha visitato tutte le cpu. Il main thread dopo aver creato i thread rimane in attesa fin tanto che tutti i thread creati non terminano.



Side Quest 1

Misurare il tempo impiegato da ogni thread e far stampare al main thread la classifica.

Potete inviare le vostre soluzioni a
matteo.federico@uniroma2.it.

Scrivete nell'oggetto della mail [SO] challenge Scheduling.

Vedremo le più interessanti (in positivo e negativo) la prossima lezione.

(Sarò anonimo nel mostrare le soluzioni)



Estratto - Esame Programmazione

Implementare una programma che riceva in input, tramite `argv[]`, il nome di un file `F`. Il programma dovrà creare il file `F` e popolare il file con lo stream proveniente da standard-input. Il programma dovrà generare anche un ulteriore processo il quale dovrà riversare il contenuto del file `F` su un altro file denominato `shadow_F`, inserendo mano a mano in `shadow_F` soltanto i byte che non erano ancora stati prelevati in precedenza da `F`. L'operazione di riversare i byte di `F` su `shadow_F` dovrà avvenire con una periodicità di 10 secondi.

Windows to Linux

Reverse File: è stato visto in Windows a lezione.
Riproduciamolo su linux!

Windows to Linux

copy File: è stato visto in Windows a lezione.

Riproduciamolo su linux, ma potenziandolo!

Dividiamo il file in chunk, preallochiamo un file della stessa dimensione.

Facciamo spawnare N thread e ogni thread prende un chunk e lo copia all'interno del file appena creato.



matching-in-file

È richiesta l'implementazione di un programma per il conteggio delle lettere. Il programma dovrà leggere un file A composto da N righe, ognuna delle quali contiene un numero M di lettere casuali. Il programma dovrà quindi generare un file B, anch'esso composto da N righe, in cui la riga i-esima riporterà il numero di occorrenze della lettera "H" presenti nella riga i-esima del file A. E' raccomandato l'utilizzo di molteplici thread per velocizzare il conteggio. **NON** ignorare problemi di sincronizzazione, ma senza usare meccanismi di sincronizzazione nativi (mutex, semaphores)
NON SONO NECESSARI!