Laboratorio di sistemi operativi
A.A. 2010/2011
Gruppo 2
Gennaro Oliva
24
Esercizi sui thread POSIX

Esercizio 38

- Si realizzi un programma che accetta come parametro sulla linea di comando due interi n e k, genera un array double di dimensione k utilizzando la funzione drand48() e ne calcola il massimo utilizzando n thread ciascuno dei quali processa k/n elementi dell'array
- Si realizzi un programma che accetta come parametro sulla linea di comando un intero n, ed un nome di file e calcoli il numero di linee contenute nel file utilizzando k thread che analizzano parti di file diverse, ma con uguale numero di caratteri
  - ./a.out 4 file.txt
    se file.txt ha 1024 byte il primo thread processa i primi 256 byte il secondo da 257 a 512 ...

Esercizio 39

- Si realizzi un programma che accetta come parametro sulla linea di comando un intero n, ed un nome di file. Il programma conta il numero di caratteri “bianchi” ('\n','\t',' ') e “non bianchi” contenuti in totale nel file utilizzando n thread che analizzano parti di file diverse, ma con uguale numero di caratteri
  - ./a.out 4 file.txt
    se file.txt ha 1024 byte il primo thread processa i primi 256 byte il secondo da 257 a 512 ...
- Ogni thread al termine dell'esecuzione aggiorna due variabili globali che memorizzano il conteggio totale dei caratteri bianchi e non e stampa il nuovo valore a video
- Si con proteggano con mutex eventuali race condition

Esercizio 40

- Si realizzi un programma C che accetta sulla linea di comando due interi n ed m e crea n thread produttori e m thread cosumatori
- I thread produttori incrementano di una unità una variabile condivisa fino ad un massimo di 20 mentre i thread consumatori la decrementano fino ad un minimo di 5
- Ogni volta che un thread ha modificato la variabile ne stampa il nuovo valore e dorme un numero di secondi variabile tra 0 e 5
- I thread utilizzano una condition variable per la sincronizzazione

Esercizio 41

- Si realizzi una versione reentrant della funzione ctime che utilizzi un area di memoria thread-specific per evitare che chiamate simultanee alla funzione possano interferire
- Si realizzi quindi un programma che accetta come unico parametro sulla linea di comando un intero n e che genera n thread che effettuano un ciclo di 10 iterazioni all'interno di ciascuna delle quali viene invocata la versione reentrant di ctime implementata e successivamente si attende un numero random di secondi tra 0 e 3