

Elaborato 4: Utilizzo Thread.

Consegna: entro il giorno dell'orale

Modalità di consegna:

- 1) Rinominare il file contenente l'elaborato con il proprio numero di matricola. Si ricorda che la consegna è individuale, pertanto ogni studente dovrà consegnare una copia dell'elaborato.
- 2) Riportare in calce al file contenente l'elaborato un commento che includa: matricola, nome e cognome, data di consegna, titolo dell'elaborato.
- 3) Fare l'upload del file su <http://amarena.sci.univr.it/>
 - a. Seguire i link: Accesso pubblico → Laboratorio Sistemi Operativi 2008 → "Nome_docente_del_corso"
 - b. A questo punto dovrete trovarvi all'interno di anonymous / Laboratorio Sistemi Operativi 2008/ Nome_docente_del_corso
 - c. Cliccare sulla freccia alla destra della voce Elaborato 4 (sotto la colonna Azione), quindi su Nuovo → Documento
 - d. Compilare i campi del form che appare inserendo il file di cui fare l'upload in "File locale", il vostro nome, cognome e n° di matricola su "Nome del documento".
 - e. Premere OK
- 4) Si ricorda inoltre che non si potranno né modificare né visualizzare i file di cui è stato fatto l'upload.
- 5) Per qualunque problema durante la sottomissione dell'elaborato contattare il docente del relativo corso (Bombieri per Informatica Multimediale, Pravadelli per Informatica).

Testo dell'elaborato

Scrivere un programma C che sfrutti le thread POSIX (Pthread) per effettuare in parallelo la somma di 12.800 numeri interi casuali. Tali numeri devono essere memorizzati, uno per riga, in un file passato al programma tramite linea di comando.

Il main deve creare una prima thread di inizializzazione il cui obiettivo consiste nel copiare i numeri dal file all'interno di 128 array (100 numeri per ogni array), e nell'inizializzare a 0 una variabile counter.

Il main, attesa la terminazione della thread di inizializzazione, deve creare 128 ulteriori thread $T[i]$, con $0 \leq i < 128$, una per ogni array, il cui obiettivo consiste innanzitutto nel calcolare la somma $S[i]$ dei 100 interi memorizzati nel corrispondente array $A[i]$. Pertanto, dopo la fase di inizializzazione il programma deve procedere come segue:

1. Ciascuna thread $T[i]$ con $0 \leq i < 128$ calcola la somma $S[i]$ degli elementi contenuti nell'array $A[i]$.
2. Ciascuna thread $T[i]$ con $0 \leq i < 64$, rimane attiva e aggiunge a $S[i]$ la somma $S[i+64]$ eseguita dalla thread $T[i+64]$. Ad esempio, la thread $T[0]$ somma a $S[0]$ il valore $S[64]$ calcolato dalla thread $T[64]$, la thread $T[1]$ somma a $S[1]$ il valore $S[65]$ calcolato dalla thread $T[65]$, etc.
3. Ciascuna thread $T[i]$ con $64 \leq i < 128$, termina la propria esecuzione.
4. I passi 2 e 3 devono esser ripetuti considerando questa volta le 64 thread rimaste attive. Pertanto, ciascuna thread $T[i]$ con $0 \leq i < 32$, rimane attiva e aggiunge a $S[i]$ la somma $S[i+32]$ eseguita dalla thread $T[i+32]$ mentre le thread $T[i]$ con $32 \leq i < 64$, terminano la propria esecuzione.
5. Il passo 4 va ripetuto ricorsivamente, fino a quando rimane una sola thread (la numero 0), che di conseguenza conterrà la somma totale dei 12.800 numeri iniziali.

Durante i passi 2-5, ciascuna thread deve incrementare di una unità la variabile condivisa counter ogni volta che viene effettuata una somma. La variabile deve essere modificata dalle thread in modo mutuamente esclusivo.

Il main attesa la terminazione di tutte le thread dovrà stampare la somma dei 128.000 numeri e il valore della variabile counter.