

C - Comunicazione tra processi

Esercizio 1

Si scriva un programma C dove N processi vengono generati dal processo padre; essi condividono un segmento di memoria in cui, il processo i -esimo scriverà il carattere 'a' + i in una parte di dimensione $\frac{N}{M}$, con offset pari a $i * \frac{N}{M}$, dove $N = 16 * M$ con $M = 6$. Ogni processo dovrà scrivere su stdout il carattere corrispondente e l'offset da cui ha iniziato a scriverlo. Il processo padre attenderà che tutti i figli abbiano concluso l'esecuzione per poi scrivere su stdout ogni carattere salvato nella memoria condivisa.

Esercizio 2

Si scriva un programma C in cui due processi si scambiano segnali, con lo scopo di simulare una partita a tennistavolo. Il primo processo scriverà su stdout una stringa corrispondente al proprio colpo ("Ping" o "Pong"), successivamente invierà un segnale al secondo processo che risponderà, e così via fino a quando non saranno stati eseguiti 10 scambi. La partita inizia con uno dei due processi scelto casualmente (il primo colpo potrebbe quindi essere un "Pong").

Threads

Esercizio 3

Si scriva un programma C in cui viene implementata la funzione `long computeFunctionValue()`, in cui viene eseguito il seguente calcolo $y = x^4 - x^3 + x^2$. Tale calcolo viene effettuato utilizzando 3 threads:

- un thread calcola x^4
- un thread calcola x^3
- un thread calcola x^2

Tale funzione crea i 3 thread, attende che i risultati siano pronti, calcola y e lo stampa su stdout.

Esercizio 4

Si scriva un programma C in cui il thread principale riempie un vettore di dimensione 9000000 con numeri random tra 0 e 9. Successivamente crea `N_THREADS` threads che si coordineranno per calcolare la somma degli elementi del vettore. La variabile contenente tale valore deve essere condivisa tra tutti i thread, i quali lavoreranno direttamente su di essa. Ciascun thread calcola la somma relativa solamente ad una porzione del vettore, di dimensione uguale per tutti, con offset diverso.