

Sistemi Operativi

Processes vs Threads ***-Specifiche Progetto-***

Flavio Bertini

flavio.bertini@unipr.it

Specifiche 1

$$\left[a_1, a_2, a_3, \dots, a_n \right]$$

- Array di n posizioni definite dall'utente
 - a_i intero random
- Calcolare la somma degli n numeri usando p processi e t thread
 - con p definito dall'utente ma compreso tra 5 e 10
 - con t definito dall'utente ma compreso tra 5 e 10
- Verificare il risultato della somma nel main
- Calcolare il tempo impiegato dai processi, dai thread e dal main
- **BONUS:** ipotizzare d'avere m array distinti e valutare il vantaggio dei due diversi approcci (processi/thread) in termini di tempo, mantenendo vive le unità di elaborazione create (processi/thread)

Specifiche 2

$$\begin{array}{ccccccc} \left[a_1, & a_2, & a_3, & \dots, & a_n \right] \\ \underbrace{\hspace{1.5cm}} & \underbrace{\hspace{1.5cm}} & \underbrace{\hspace{1.5cm}} & \underbrace{\hspace{1.5cm}} & \underbrace{\hspace{1.5cm}} \\ & p_1 & p_2 & \dots & p_p \\ & t_1 & t_2 & \dots & t_p \end{array}$$

- Suddividere la computazione tra i p/t processi/thread
- Usare le pipe per la comunicazione dei dati tra i processi
- Usare meccanismi di lock visti per sincronizzare i thread nella generazione del risultato finale
- Nel calcolo dei tempi mantenere separate le fasi di creazione delle unità di elaborazione dall'effettivo calcolo

Consegna

- Generare un unico archivio
cognome_nome_matricola.tar.gz contenente i 2 programmi in C relativi alle due soluzioni
- Inviare l'archivio al docente come allegato mail entro le ore 23:59 del giorno (*giorno_appello - 7 giorni*)
- Sono ammessi gruppi di massimo 2 persone (ai quali dovrà corrispondere una sola consegna); in tal caso indicare nel corpo della mail i rispettivi nomi e le matricole
- La consegna resterà valida per tutti gli appelli dell'anno