

Esercitazione [04]

Riepilogo su Processi, Thread e Semafori

Leonardo Aniello – aniello@dis.uniroma1.it

Daniele Cono D'Elia – delia@dis.uniroma1.it

Giuseppe Laurenza – laurenza@dis.uniroma1.it

Federico Lombardi – lombardi@dis.uniroma1.it

Sistemi di Calcolo - Secondo modulo (SC2)

Programmazione dei Sistemi di Calcolo Multi-Nodo

Corso di Laurea in Ingegneria Informatica e Automatica

A.A. 2016-2017

Esercizio proposto (1/4)

[Secondo esercizio di riepilogo]

- Sviluppare un'applicazione in C con questa semantica
 - Il processo “main” crea M processi figlio tramite fork
 - Ogni processo figlio è identificato da un `child_id`
 - L'attività dei processi figlio consiste nel lanciare N thread, attenderne la fine, e sincronizzarsi per accedere ad una risorsa condivisa per stampare una variabile (vedi prossimo punto)
 - Ogni thread è identificato da un `thread_id`
 - La risorsa condivisa può essere acceduta al più da `NUM_RESOURCES` per volta
 - L'attività dei thread consiste nel calcolare il prodotto `thread_id*child_id` ed aggiungerlo ad una variabile contatore condivisa tra tutti i thread del processo (da accedere in mutua esclusione)

Esercizio proposto (2/4)

- Processo «main»
 - Crea un counting semaphore per proteggere l'accesso alla risorsa condivisa tra tutti i processi
 - Crea M processi figlio
 - Attende il termine di tutti i processi figlio
 - Rilascia le risorse allocate

Esercizio proposto (3/4)

- Processo figlio
 - Crea un semaforo binario per proteggere l'accesso in mutua esclusione alla variabile condivisa tra tutti i suoi thread
 - Lancia N thread
 - Attende il termine di tutti i thread
 - Si sincronizza con gli altri processi per accedere alla risorsa condivisa
 - Una volta ottenuto l'accesso, stampa il valore della variabile condivisa tra i thread

Esercizio proposto (4/4)

- Thread di un processo figlio
 - Si sincronizza con gli altri thread dello suo stesso processo per accedere in sezione critica
 - Una volta in sezione critica, calcola il prodotto tra `child_id*thread_id` e lo aggiunge alla variabile condivisa