

Esercitazione 9

POSIX threads, gestione della mutua esclusione. Per gli esercizi proposti si richiede di realizzare un Makefile.

Esercizio 1

Scrivere un programma C con due threads, un produttore (P) ed un consumatore (C). Il thread P genera, uno alla volta, una sequenza di numeri inserendoli in un buffer di 1 posizione condiviso con il thread C. Il thread consumatore estrae i numeri dal buffer e li stampa sullo standard output. Se il buffer e' pieno P attende che C consumi il dato, analogamente se il buffer e' vuoto C attende che P produca un valore da consumare.

Esercizio 2

N filosofi siedono ad una tavola con un piatto di spaghetti davanti ed una forchetta alla loro destra ed una alla loro sinistra. Per mangiare gli spaghetti un filosofo ha bisogno di entrambe le forchette vicine. Ogni filosofo è impegnato ininterrottamente in una sequenza di 3 attività: meditare, cercare di acquisire le forchette e mangiare. Scrivere un programma C che attivi N thread filosofi che eseguono il ciclo descritto per 100 volte. La meditazione e la fase in cui il filosofo mangia deve essere implementata con un ritardo variabile (usare ad esempio la chiamata di sistema `nanosleep` e la funzione `rand_r()`).

Esercizio 3

Scrivere un programma C in cui si attivano M thread produttori ed N thread consumatori che condividono una coda (di lunghezza "infinita"). Il programma accetta come argomento anche un intero N che corrisponde al numero totale di messaggi che i produttori producono concorrentemente nella coda (ogni produttore produce N/M messaggi se M divide N). I consumatori leggono i messaggi dalla coda in un ciclo infinito fino a quando non ricevono un messaggio speciale che li fa terminare. Implementare la coda concorrente, ed il protocollo di terminazione senza usare la cancellazione esplicita dei threads.