

Esercitazione [04]

Riepilogo sulla sincronizzazione

Riccardo Lazzeretti – lazzeretti@diag.uniroma1.it

Sistemi di Calcolo 2

Programmazione dei Sistemi di Calcolo Multi-Nodo

Corso di Laurea in Ingegneria Informatica e Automatica

A.A. 2020-2021

Esercizio proposto (1/4)

[Esercizio di riepilogo su quanto visto finora in laboratorio]

- Sviluppare un'applicazione in C con questa semantica
 - Il processo «main» crea N processi figlio tramite fork
 - Tutti i processi figlio si sincronizzano per iniziare la loro attività, avviata dal processo «main»
 - L'attività dei processi figlio consiste nel lanciare M thread per volta
 - La sezione critica di ciascun thread consiste nello scrivere in append su un file l'identità del processo scrivente
 - Passati T secondi, il processo «main» deve notificare i processi figlio di cessare la loro attività e terminare (usare `sem_getvalue()`)
 - Prima di terminare, un processo deve attendere la fine dei thread attualmente in esecuzione
 - Infine, il processo «main» deve identificare il processo che ha effettuato più accessi in sezione critica

Esercizio proposto (2/4)

- Processo «main»
 - Crea N processi figlio
 - Notifica gli N processi figlio di avviare la loro attività
 - Attende T secondi
 - Notifica gli N processi figlio di cessare la loro attività e terminare
 - Attende la terminazione degli N processi figlio
 - Identifica il processo che ha acceduto in sezione critica più volte (usare la funzione `parseOutput()`)
 - Termina

Esercizio proposto (3/4)

- Processo figlio
 - Attende la notifica di avvio dal processo «main»
 - Ciclo
 - Lancia M thread
 - Attende il termine degli M thread
 - Verifica se il processo «main» ha notificato di cessare l'attività
 - In caso positivo, esce dal ciclo
 - Termina

Esercizio proposto (4/4)

- Thread di un processo figlio
 - Richiede l'accesso in sezione critica
 - Una volta in sezione critica
 - Apre il file in append
 - Scrive l'identità del processo figlio
 - Chiude il file
 - Esce dalla sezione critica
 - Termina