

Prova pratica del 3/3/2016
Durata della prova: 150 minuti

Cognome Nome Matr.

Lo studente legga attentamente il testo e produca il programma, il makefile, ed i casi di test necessari per dimostrarne il funzionamento. La mancata compilazione dell'elaborato, la compilazione con errori o l'esecuzione errata del programma daranno luogo alla valutazione come **prova non superata**. Ricordarsi di indicare Nome, Cognome e matricola su questo stesso foglio, che dovrà essere in ogni caso consegnato alla Commissione. Al termine della prova lo studente dovrà fare verificare il funzionamento del programma ad un membro della Commissione.

Testo della prova

Si realizzi in linguaggio C/C++ una applicazione **multiprocesso** che simuli delle elaborazioni su una matrice. La matrice dovrà essere allocata in una **shared memory UNIX**, e la sincronizzazione dovrà essere basata su **semafori UNIX**. La applicazione dovrà creare i seguenti processi:

- Un processo **generatore**, che periodicamente (ogni 3 secondi) aggiorna il contenuto della matrice con valori casuali (scelti tra 0 e 9). L'operazione di aggiornamento dovrà durare 1 secondo, da simulare mediante la primitiva `sleep()`.
- Un processo **elaboratore**, che periodicamente (ogni secondo) legge il contenuto della matrice, moltiplica per 2 il valore di ogni elemento, e stampa a video il risultato dei prodotti.
- Due processi **analizzatore**, che periodicamente (ogni 2 secondi) leggono il contenuto della matrice, calcolano il valore medio degli elementi, e stampano a video il risultato.

I processi dovranno sincronizzarsi usando lo schema dei **lettori-scrittori, con starvation di entrambi**. I processi **elaboratore** e **analizzatore** dovranno fungere da lettori, il processo **generatore** da scrittore.

Il codice dei tre tipi processi dovrà risiedere in **tre eseguibili distinti**. Inoltre, è richiesto un ulteriore eseguibile "principale" per allocare le risorse e per avviare i processi. Il programma principale dovrà allocare nella memoria condivisa una **matrice quadrata NxN** su cui i processi figli svolgeranno le operazioni. Il valore N dovrà essere scelto casualmente (compreso tra 3 e 5) dal programma principale a tempo di esecuzione, e dovrà essere trasmesso ai processi figli, con modalità a scelta dello studente. Il programma principale dovrà forzare la terminazione dei processi figli dopo 15 secondi, attraverso una ulteriore variabile condivisa, inizializzata a 0, e su cui scriverà il valore 1 per indicare ai figli di terminare. Al termine di tutti i processi, il programma principale dovrà de-allocare le risorse.