

Si realizzi in linguaggio C/C++ una applicazione **multiprocesso** che simuli delle elaborazioni su una stringa. La stringa 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 stringa con valori casuali (scelti tra i caratteri 'a' e 'z'; è possibile utilizzare `'a' + (rand() % 26)`). 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 stringa, converte in maiuscolo ogni carattere della stringa (basta sottrarre al carattere ASCII il valore intero 32), e stampa a video il risultato.
- Due processi **analizzatore**, che periodicamente (ogni 2 secondi) leggono il contenuto della stringa, ne invertono il contenuto, e stampano a video il risultato.

I processi dovranno sincronizzarsi usando lo schema dei **lettori-scrittori, con starvation dei soli scrittori**. 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" che allochi le risorse necessarie e avvii i processi tramite `exec()`. Il programma principale dovrà allocare nella memoria condivisa una **stringa di lunghezza N** su cui i processi figli svolgeranno le operazioni. Il valore N dovrà essere scelto casualmente (compreso tra 10 e 15) dal programma principale a tempo di esecuzione, e dovrà essere trasmesso ai processi figli, con modalità a scelta dello studente. Al termine di tutti i processi, il programma principale dovrà de-allocare le risorse.