Università degli Studi di Napoli Federico II
Corso di Laurea in Ingegneria Informatica
Esame di Sistemi Operativi
Proff. Cinque, Cotroneo, Pietrantuono

Prova pratica del 20/12/2019
Durata della prova: 150 minuti

Cognome ....…….................. Nome ...….............................. Matr. .... Docente ....

°°

*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 su questo foglio nome, cognome, matricola, e il docente di riferimento. Al termine della prova, lo studente dovrà consegnare questo foglio alla Commissione, e fare verificare il funzionamento del programma ad un membro della Commissione.*

°°

Testo della prova

Si realizzi in linguaggio C/C++ un'applicazione **multiprocesso** per parallelizzare dei calcoli matematici, basata su **code di messaggi**, costrutto **monitor signal and wait** e sullo **schema lettori-scrittori con starvation di entrambi.**

```
typedef struct {
      int risultati[3];
      /* ... aggiungere altre
            variabili per la
            sincronizzazione ...
      */
} MonitorRisultati;
```

```
void inserisci_risultato(MonitorRisultati * mr, int risultato, int operazione);
int leggi_risultati(MonitorRisultati * mr);
```

Il programma dovrà prevedere tre gruppi di processi. Il primo gruppo (3 processi) dovrà **generare gli operandi** (4 volte ogni processo), che sono valori casuali **(float)** tra 1 e 20. Il valore andrà inviato asincronamente alla mailbox di operandi. Se la coda non ha spazio, il processo chiamante viene posto in attesa. Ogni processo genera operandi per una operazione diversa: somma, prodotto e divisione.
Il secondo gruppo (3 processi) dovrà **prelevare gli operandi dalla mailbox** (4 chiamate per ogni processo). Ogni processo del gruppo è associato a una operazione matematica tra le tre elencate sopra (somma, prodotto e divisione), e preleva dalla mailbox **solo** gli operandi per quella operazione. Il processo chiamante viene posto in attesa se non ci sono operandi da prelevare.

Ogni due operandi prelevati, i processi del secondo gruppo dovranno **effettuare l'operazione** a cui sono associati sui due operandi, attendere 1 secondo, ed inserire il risultato nel monitor chiamando il metodo `inserisci_risultato()` (2 chiamate per ogni processo), che colloca il valore nella struttura dati `risultati` secondo uno schema **lettori-scrittori** con starvation di entrambi.

Il terzo gruppo (3 processi) deve **leggere i tre risultati** dal monitor, chiamando ripetutamente (6 volte in totale) il metodo `leggi_risultati()` , e successivamente stampando i risultati a video.