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

Prova pratica del 22/01/2015
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++ un'applicazione **multiprocesso** secondo il seguente schema. Tre processi, detti **Server**, gestiscono una **risorsa condivisa** (rappresentata da una **coppia di variabili intere**) collocata in una **shared memory UNIX**. Su richiesta di un secondo gruppo di tre processi, detti **Client**, i Server aggiornano la risorsa sovrascrivendola con una nuova coppia di valori. Le richieste di modifica dovranno essere inviate tramite una **coda di messaggi UNIX**. Il messaggio con la richiesta dovrà contenere una nuova coppia di valori da sovrascrivere. I Server dovranno modificare e stampare a video il valore della risorsa in **mutua esclusione**.

Quando i Server ricevono una richiesta di modifica, essi attendono un secondo prima di procedere con la modifica. I processi Client attendono un tempo casuale tra 1 e 2 secondi tra l'invio di una richiesta e la richiesta successiva, e scelgono in maniera casuale una coppia di valori tra 0 e 9 da inserire nella richiesta. Ogni Client dovrà generare 4 richieste, per poi terminare. Ogni Server dovrà servire 4 richieste, per poi terminare. Occorre sviluppare **tre eseguibili distinti**, di cui uno sarà eseguito dai processi Client, uno sarà eseguito dai processi Server, e il terzo sarà eseguito dall'utente per avviare l'esecuzione dell'applicazione. Il terzo eseguibile dovrà avviare tutti i processi Client e Server utilizzando le primitive `fork()` ed `exec()`, attenderne la terminazione, e terminare a sua volta.