Università di Napoli Federico II
Corso di Laurea in Ingegneria Informatica e Elettronica
Esame di Sistemi Operativi
Proff. De Carlni, Cotroneo, Cinque

Prova pratica
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** per la simulazione di un servizio meteo. L'applicazione sia costituita da 11 processi, di cui un processo, denominato nel seguito M, rappresenti il servizio di meteo, e gli altri 10 processi rappresentino ipotetici utenti del servizio.

Il processo M detiene le informazioni di meteo nella seguente struttura **meteo**:

```
typedef struct {
      int temperatura;
      unsigned int umidita;
      unsigned short pioggia;
} meteo;
```

dove la variabile "temperatura" è un intero compreso tra -50 e +50, la variabile "umidita" è espressa in percentuale tra 0 e 100, e la variabile "pioggia" indica se il tempo è piovoso (1) o non piovoso (0).

Il processo M genera in maniera casuale[1] ogni 2 secondi le informazioni meteo, memorizzandole nella struttura. M termina dopo aver effettuato 20 scritture. La scrittura deve avvenire in mutua esclusione.

Ognuno dei 10 processi utente effettuano un'operazione di lettura ogni secondo dalla struttura "meteo", stampando a video le informazioni lette. I processi utente terminano dopo aver effettuato 10 letture. Due o più processi utente possono leggere contemporaneamente dalla struttura, se non occupata da M.

La struttura "meteo" deve essere memorizzata in un segmento di memoria condivisa, e l'accesso a tale struttura da parte di M e dei processi utente deve essere disciplinato attraverso un **monitor**.

Il processo M e i 10 processi utente sono generati dal programma principale attraverso le primitiva fork. Una volta generati i processi, il programma principale ne attende la terminazione prima di eliminare eventuali risorse condivise.

---

[1] La generazione casuale può essere implementata con la funzione rand() di stdlib.h; ad esempio: int umidita = rand() % 101 rand() richiede che venga generato un seme dei numeri casuali attraverso la funzione srand(time(NULL)).