

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

**Prova pratica del 18/02/2011**  
**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 di prenotazione posti in un teatro. L'applicazione è costituita da 51 processi, 50 *clienti* e 1 *visualizzatore*. I posti nel teatro sono rappresentati da un vettore di 80 elementi, ognuno contenente la seguente struttura **posto**:

```
typedef struct {
    unsigned int id_cliente;
    unsigned int stato;
} posto;
```

dove `id_cliente` è il PID del cliente che ha prenotato il posto (inizialmente 0), e `stato` indica se il posto è *libero* (0), *occupato* (1), o *in\_aggiornamento* (2) da parte di un altro cliente. Una variabile globale `disponibilita` indica il numero di posti ancora presenti in teatro (inizialmente 80).

Ognuno dei clienti dapprima attende un tempo casuale da 0 a 5 secondi (con una *sleep*), poi effettua un tentativo di prenotazione da 1 a 4 posti (il numero di posti e i secondi di attesa possono essere generati in maniera casuale con la funzione `rand()`<sup>1</sup>). Ad ogni tentativo di prenotazione, il client stabilisce se c'è spazio, in base alla variabile `disponibilita`, altrimenti termina subito con un messaggio di "disponibilità esaurita". Se c'è disponibilità, pone in aggiornamento i primi posti liberi che trova, li occupa, assegnando ai posti il proprio PID in `id_cliente` (si simuli con una *sleep* un tempo di 1 secondo per occupare i posti), ed aggiorna la `disponibilita`.

Il processo visualizzatore stampa a video, ogni secondo, la situazione dei posti nel teatro, scrivendo su ogni riga il numero di posto, lo stato, ed eventualmente il cliente che ha occupato il posto.

Il vettore dei posti e la variabile `disponibilita` sono variabili condivise, e l'accesso a tali variabili da parte dei processi deve essere disciplinato attraverso l'uso di **Semafori**.

I processi clienti sono generati da un programma principale attraverso la primitiva `fork()`. Una volta generati i processi, il programma principale ne attende la terminazione e termina a sua volta. Il processo visualizzatore venga invece implementato in un programma principale a parte.

---

<sup>1</sup> La generazione casuale può essere implementata con la funzione `rand()` di `stdlib.h`; ad esempio: `int num = rand() % 4 + 1`  
`rand()` richiede che venga generato un seme dei numeri casuali attraverso la funzione `srand(time(NULL))`.