

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 24/06/2011**  
**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 algoritmo di schedulazione dei dischi. L'applicazione deve essere costituita da un processo **Schedulatore** e da un gruppo di 5 processi **Utente**. Ogni processo Utente genera 5 richieste di scrittura verso il disco, che devono essere collocate in una **coda circolare di 10 elementi allocata in una memoria condivisa**, per poi terminare la sua esecuzione. Se la coda è piena, il processo Utente deve mettersi in attesa che vi sia una posizione disponibile. Una richiesta deve contenere un valore da scrivere sul disco (si utilizzi il PID del processo) e la posizione in cui scrivere il valore (un valore intero casuale tra 0 e 19):

```
typedef struct {  
    unsigned int posizione;  
    pid_t processo;  
} richiesta;
```

Il processo Schedulatore preleva le richieste dalla testa della coda circolare (ossia applicando una politica FIFO). Per simulare la scrittura su disco, lo Schedulatore attende per un intervallo di tempo (tramite la primitiva `sleep()`) di durata  $t_i = |p_i - p_{i-1}|$  secondi, dove  $p_i$  rappresenta la posizione sul disco della  $i$ -esima scrittura, assumendo  $p_0=0$ . Dopo aver atteso  $t_i$  secondi, lo Schedulatore sovrascrive il valore della richiesta alla posizione  $p_i$  di un array rappresentante il disco (da allocare come variabile automatica). Lo Schedulatore termina dopo aver servito 25 richieste provenienti dai processi Utente.

L'accesso da parte dei processi alla coda circolare e ai relativi puntatori "testa" e "coda" deve essere disciplinato attraverso il costrutto **Monitor di Hoare**.

I processi Utente e Schedulatore sono generati da un unico programma principale attraverso la primitiva `fork()`. Una volta generati i processi, il programma principale ne attende la terminazione e termina a sua volta.

