

# Prova pratica di Calcolatori Elettronici (nucleo v6.\*)

C.d.L. in Ingegneria Informatica, Ordinamento DM 270

26 giugno 2010

1. Definiamo un *monitor* come un oggetto che può essere posseduto da un solo processo alla volta e a cui è associata una *variabile di condizione*. Il possessore del monitor può eseguire le operazioni di *attesa* e *notifica* sulla variabile di condizione associata. L'operazione di attesa rilascia il monitor, sospende il processo fino a quando un altro processo esegue l'operazione di notifica sulla stessa variabile di condizione, quindi riacquiesce il monitor. L'operazione di notifica risveglia uno dei processi in attesa sulla variabile di condizione, se ve ne sono, altrimenti non fa niente.

Per realizzare i monitor definiamo la seguente struttura (file `sistema.cpp`):

```
struct des_monitor {
    natl owner;
    natl mutex;
    natl cond;
    int num_waiting;
};
```

Il campo `owner` contiene l'id del processo che possiede il monitor (0 se nessun processo possiede attualmente il monitor). I campi `mutex` e `cond` sono due indici di semafori. Il campo `num_waiting` tiene conto del numero di processi in attesa sulla variabile di condizione.

Le seguenti primitive, accessibili dal livello utente, operano sui monitor (nei casi di errore, abortiscono il processo chiamante):

- `natl monitor_ini()` (già realizzata): inizializza un nuovo monitor, con i campi `owner` e `num_waiting` entrambi a 0, `mutex` con l'indice di un semaforo di valore iniziale 1, `cond` con l'indice di un semaforo di valore iniziale 0. Restituisce l'identificatore del nuovo monitor. Se non è possibile creare un nuovo monitor, restituisce `0xFFFFFFFF`.
- `void monitor_enter(natl mon)` (già realizzata): tenta di impadronirsi del monitor di identificatore `mon`. Se il monitor appartiene già a qualche altro processo, sospende il processo in attesa che il monitor venga prima rilasciato. È un errore tentare di impadronirsi di un monitor che si possiede già.
- `void monitor_leave(natl mon)` (già realizzata): rilascia il monitor di identificatore `mon`. È un errore tentare di rilasciare un monitor che non si possiede.
- `void monitor_wait(natl mon)`: si pone in attesa sulla variabile di condizione associata al monitor di identificatore `mon`. È un errore tentare di porsi in attesa su una variabile di condizione associata ad un monitor che non si possiede.
- `void monitor_notify(natl mon)`: esegue una notifica sulla variabile di condizione associata al monitor di identificatore `mon`. È un errore tentare di eseguire una notifica su una variabile di condizione associata ad un monitor che non si possiede.

Modificare i file `sistema.cpp` e `sistema.s` in modo da realizzare le primitive mancanti.