

Prova pratica di Calcolatori Elettronici (nucleo v6.*)

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

5 febbraio 2011

1. Definiamo un *rw* come un oggetto su cui i processi possono leggere o scrivere rispettando le seguenti condizioni:

1. più processi possono leggere contemporaneamente, purchè nessun processo stia scrivendo;
2. un solo processo alla volta può scrivere.

Supporteremo anche che ci sia un massimo (`MAX_RW_READERS`) al numero di processi che possono leggere contemporaneamente.

Per leggere o scrivere su un *rw*, i processi devono prima acquisire il diritto di lettura o scrittura, quindi devono rilasciare tale diritto quando hanno terminato.

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

```
struct des_rw {
    natl readers[MAX_RW_READERS];
    natl writer;
    natl nreaders;
    des_proc* w_readers;
    des_proc* w_writers;
};
```

Il campo `readers` memorizza gli *id* degli eventuali processi che hanno acquisito il diritto di lettura e non lo hanno ancora rilasciato. Il campo `nreaders` memorizza il numero di tali processi. Il campo `writer` memorizza l'*id* dell'eventuale processo che ha acquisito il diritto di scrittura e non lo ha ancora rilasciato. La lista `w_readers` contiene i processi in attesa di acquisire il diritto di lettura. La lista `w_writers` contiene i processi in attesa di acquisire il diritto di scrittura.

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

- `natl rw_init()` (già realizzata): inizializza un nuovo *rw* e ne restituisce l'identificatore. Se non è possibile creare un nuovo *rw* restituisce `0xFFFFFFFF`.
- `void rw_acq_write(natl rw)` (già realizzata): tenta di acquisire il diritto di scrittura sul *rw* di identificatore `rw`. Se ci sono già processi che hanno acquisito un diritto e non lo hanno ancora rilasciato, sospende il processo in attesa che le condizioni permettano l'acquisizione del diritto di scrittura.
- `void rw_acq_read(natl rw)` (già realizzata): tenta di acquisire il diritto di lettura sul *rw* di identificatore `rw`. Se c'è un processo che ha acquisito il diritto di scrittura e non lo ha ancora rilasciato, oppure se ci sono già `MAX_RW_READERS` processi che hanno acquisito il diritto di lettura e non lo hanno ancora rilasciato, sospende il processo in attesa che le condizioni permettano l'acquisizione del diritto di lettura.

- `void rw_rel_write(natl rw)`: Rilascia il diritto di scrittura sul `rw` di identificatore `rw`. Se vi sono processi in attesa di acquisire un diritto di lettura o scrittura, lo concede dando precedenza ai lettori e cercando di concedere il diritto a più processi possibile, nel rispetto delle condizioni.
- `void rw_rel_read(natl rw)`: Rilascia il diritto di lettura sul `rw` di identificatore `rw`. Se vi sono processi in attesa di acquisire un diritto di lettura o scrittura, lo concede dando precedenza agli scrittori e cercando di concedere il diritto a più processi possibile, nel rispetto delle condizioni.

È sempre un errore tentare di acquisire un diritto se se ne possiede già uno, o tentare di rilasciare un diritto che non si possiede.

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