

Prova pratica di Calcolatori Elettronici (nucleo v6.*)

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

11 giugno 2014

1. Una *barriera* è un meccanismo di sincronizzazione tra processi con la seguente proprietà: ogni processo che arriva alla barriera si deve bloccare fino a quando non sono arrivati alla barriera tutti i processi.

Vogliamo realizzare il meccanismo della barriera nel nucleo, ma solo tra i processi che si sono preventivamente registrati. Per questo scopo aggiungiamo le seguenti primitive:

- `void reg()` (da realizzare): registra il processo corrente sulla barriera. È un errore se un processo tenta di registrarsi due volte, oppure se tenta di registrarsi dopo che un qualunque processo è già giunto alla barriera.
- `void barrier()` (da realizzare): informa il sistema che il processo corrente è giunto alla barriera. È un errore se un processo invoca questa primitiva senza essersi registrato.

Le primitive abortiscono il processo chiamante in case di errore e tengono conto della priorità tra i processi.

Modificare i file `sistema.cpp` e `sistema.s` in modo da realizzare le primitive appena descritte. Può essere necessario definire nuove strutture dati e aggiungere campi al descrittore di processo.