

Quesito 1. Si definisca la situazione denominata *race condition* nel modo più formale e rigoroso possibile, specificando le circostanze che la inducono. Si indichino gli effetti che essa produce, discutendo se essi siano dannosi e pertanto da evitare, fornendo argomentazioni giustificate a sostegno. Si fornisca un esempio concreto di *race condition*, espresso in un linguaggio a piacere. Si indichi almeno una tecnica di programmazione per la rimozione od il trattamento del problema e se ne valuti l'efficacia.

Quesito 2. Si specifichi la relazione che intercorre tra i paradigmi detti *avoidance synchronisation* e *condition synchronisation* e la mutua esclusione. Si raffrontino i 2 paradigmi tra di loro rispetto a:

- potere espressivo conferito al modello di concorrenza
- possibili costrutti del linguaggio di implementazione e relativo costo realizzativo.

Si forniscano esempi dell'uno e dell'altro paradigma espressi in un linguaggio a piacere, indicando con chiarezza ove il linguaggio utilizzato non ne consenta la piena espressione.

Quesito 3. Il verificarsi di una condizione di *inversione di priorità* è considerato evento indesiderabile per i programmi concorrenti i cui processi abbiano attributi di priorità ed operino in regime di ordinamento a priorità. Si definisca tale condizione nel modo più formale e rigoroso possibile, spiegando perchè essa è ritenuta indesiderabile e se essa sia del tutto eliminabile.

Una tecnica proposta per il trattamento del problema è detta *Base Priority Inheritance*. Essa attribuisce al processo P in possesso della risorsa condivisa la priorità di ogni nuovo processo Q che faccia richiesta di accesso ove $Pr(Q) > Pr(P)$, per poi ripristinare la priorità originale al rilascio della risorsa.

Si confronti l'effetto e l'efficacia di tale tecnica con quella dell'*Immediate Priority Ceiling Inheritance* vista a lezione, applicando entrambe ad un esempio concreto ed estrapolandone conclusioni generali.

Quesito 4. Uno dei classici paradigmi architetturali (*design pattern*) per la strutturazione concorrente di un servente utilizza un processo *dispatcher* in ricezione delle richieste di servizio in ingresso, ed un insieme di processi esecutori (*worker*) che prendano in carico l'esecuzione delle attività richieste.

Si raffrontino i vantaggi e gli svantaggi derivanti dall'utilizzo del modello a risorse protette con quelli del modello a *rendez-vous* per realizzare il paradigma di cooperazione (i.e. comunicazione e condivisione dati) all'interno di un tale servente.