

Programmazione concorrente

Si consideri il codice dato, che implementa un programma multi-thread in cui il main thread aggiorna periodicamente un dato condiviso, mentre i thread UtenteDato alternano elaborazioni locali a letture del dato condiviso.

Si chiede di modificare il codice dato in modo che i thread UtenteDato leggano il dato condiviso quando questo viene modificato. A questo scopo, ogni volta che modifica il dato condiviso, il thread main manda un segnale ai thread UtenteDato, che reagiscono leggendo il dato condiviso.

Evitare i problemi tipici della programmazione concorrente (corse critiche, starvation, deadlock, ecc.).

Si ricorda che bisogna caricare il file .java (NON i .class) in un unico file zip.

Programmazione distribuita con socket

Si consideri il codice dato, che implementa un sistema client-server in cui il server gestisce dei dati e il client chiede di scrivere o leggere tali dati. Dopo aver ricevuto una richiesta “Trova”, il server provvede ad inviare il risultato al client non appena il dato cercato è disponibile. La lettura della risposta da parte del client potrebbe dunque risultare sospensiva.

Si chiede di modificare il codice in modo che il client dopo aver inviato la richiesta “Trova” non aspetti la risposta, ma prosegua con l’elaborazione. Il comportamento del server non deve essere modificato. Il client invece deve poter ricevere la risposta in qualunque momento durante l’elaborazione successiva all’invio della richiesta “Trova”.

Evitare i problemi tipici della programmazione concorrente (corse critiche, starvation, deadlock, ecc.).

Si ricorda che bisogna caricare il file .java (NON i .class) in un unico file zip.

Programmazione distribuita con RMI

Si consideri il codice dato, che implementa un sistema client-server in cui il server gestisce dei dati e il client chiede di scrivere o leggere tali dati.

Si chiede di modificare il codice in modo che l’operazione trovaDato termini immediatamente. Dopo aver ricevuto una chiamata di metodo trovaDato, il server provvede ad inviare una notifica al client non appena il dato cercato è disponibile. In pratica bisogna implementare il meccanismo “call-back”.

Evitare i problemi tipici della programmazione concorrente (corse critiche, starvation, deadlock, ecc.).

Si ricorda che bisogna caricare il file .java (NON i .class) in un unico file zip.