

Progetto III - Programmazione di sistema

Istruzioni e testi degli esercizi
per il corso di
Laboratorio di Sistemi Operativi
A.A. 2010-2011

Modalità di superamento della terza parte del corso

Per superare la terza parte del corso di Laboratorio di Sistemi Operativi, è necessario svolgere, individualmente o in un gruppo di due persone, il progetto relativo alla programmazione di sistema che consiste in un esercizio a scelta tra i due proposti nelle successive slide.

Ogni esercizio richiede la progettazione e l'implementazione di due programmi C, opportunamente commentati. Ogni programma dovrà occuparsi della gestione degli errori (es., input errato da parte dell'utente) e produrre output comprensibile (es., opportune richieste all'utente). L'esercizio dovrà essere descritto in una breve relazione in formato OpenDocument o PDF che illustri le caratteristiche salienti dei programmi proposti ed un esempio di funzionamento.

Il progetto dovrà essere inviato al docente del corso tramite e-mail entro il 23 settembre 2011 (entro il 31 agosto 2011 se si desidera registrare l'esame nella sessione di settembre). Dopo tale data sarà necessario seguire le istruzioni del corso per l'anno successivo. Il voto in trentesimi (più eventuale lode per i progetti che si saranno contraddistinti) peserà 8/18 sul voto complessivo del laboratorio, che, a sua volta, contribuirà, in base ai crediti, alla formulazione del voto finale dell'esame di Sistemi Operativi e Laboratorio.

Esercizio 1: tic-tac-toe_x2 client-server

Si progettino ed implementino due programmi C, `tic-tac-toe_x2_server` e `tic-tac-toe_x2_client`, che utilizzino le chiamate di sistema per permettere di giocare a tic-tac-toe a due persone collegate allo stesso sistema da diverse console (reali o virtuali).

Il programma client dovrà richiedere il nome all'utente e disegnare una griglia 3x3 indicando le lettere a,b,c sulle righe e i numeri 1,2,3 sulle colonne. Il giocatore che comincia verrà scelto a caso dal server.

Il programma server gestirà poi turni, chiedendo di volta in volta, al client dell'utente corrente di richiedergli dove posizionare il suo simbolo mediante le coordinate sopra indicate (es., l'utente dovrà digitare a3 per inserire il suo simbolo nella terza colonna della prima riga). Il client controllerà l'input, lo invierà al server e aggiornerà la griglia con i simboli. Il gioco proseguirà fino alla vittoria di uno dei due giocatori o fino al riempimento della griglia. Nel primo caso verrà comunicato il vincitore, nel secondo la situazione di parità, dopodiché i due client si scollegheranno dal server.

Il server deve essere in grado di gestire più partite contemporaneamente: ogni due client connessi, il server deve avviare e gestire una nuova partita indipendente.

Esercizio 2: tic-tac-toe_ai client-server

Si progettino ed implementino due programmi C, `tic-tac-toe_ai_server` e `tic-tac-toe_ai_client`, che utilizzino le chiamate di sistema per permettere agli utilizzatori del client di giocare a tic-tac-toe da remoto contro un'intelligenza artificiale (AI) gestita dal server.

Il programma client dovrà disegnare una griglia 3x3 indicando le lettere a,b,c sulle righe e i numeri 1,2,3 sulle colonne e si collegherà al server, che sceglierà a caso se comincia l'utente o l'AI e gestirà i turni. Al suo turno, il giocatore verrà invitato a specificare dove posizionare il suo simbolo mediante le coordinate sopra indicate (es., l'utente dovrà digitare a3 per inserire il suo simbolo nella terza colonna della prima riga), mentre al turno dell'AI il server sceglierà la posizione che gli permette di vincere, quella che blocca una vincita dell'utente, o quella che gli possa permettere di vincere con le mosse successive. Dopo ogni turno, il programma client dovrà aggiornare la griglia disegnando i simboli nelle posizioni scelte. Il gioco proseguirà fino alla vittoria dell'utente o dell'AI, o fino al riempimento della griglia. Nel primo caso verrà comunicato il vincitore, nel secondo la situazione di parità.

Il server deve essere in grado di gestire più partite contemporaneamente: per ogni client connesso, il server deve avviare e gestire una nuova partita indipendente.