ANDREA RIGON

VR486033

UNIVERSITÀ di VERONA

RELAZIONE
ELABORATO SYSTEM CALL 2023 / 2024

Scrivere un'applicazione in linguaggio "C" che sfruttando le system call SYSTEMV viste a lezione implementi il gioco "TRIS", il gioco si basa sulle regole del classico gioco da tavolo, implementato con alcune varianti e in grado di funzionare in ambiente UNIX/LINUX da 2 utenti.
Il gioco si svolge tra due giocatori, su un campo rettangolare di dimensione 3x3 dove ogni giocatore a turno indica le coordinate dove inserire il proprio simbolo che andrà a posizionarsi nella posizione indicata. Il giocatore che avrà vinto sarà il primo giocatore che sarà riuscito ad allineare 3 propri simboli. Il tris potrà avvenire senza interruzione in orizzontale, verticale o diagonale.

SPECIFICHE

L'elaborato si compone in 3 parti:

1) **TriServer** : Si occupa di predisporre l'area di gioco in termini di memoria , condivisa , semafori e timeout. La riga di esecuzione ad esempio sarà:

./TriServer 10 X O

Nell'ordine i parametri saranno: timeout , simbolo giocare 1 e simbolo giocatore 2. Il server gestirà errori di esecuzione dovuti alla presenza precedente di campi di gioco (memoria condivisa e/o semafori), dovrà inoltre terminare in modo corretto e coerente qualora venga premuto due volte di seguito il comando CTRL-C indicando alla prima pressione che una seconda pressione comporta la terminazione del gioco È compito del server "arbitrare" la partita, ovvero sarà il server a decidere dopo ogni giocata se il giocatore che ha giocato ha vinto o meno la partita, il server dovrà segnalare di conseguenza ai client anche se hanno vinto o meno. Il server dovrà inoltre notificare ai client, quando non sono più possibili inserimenti di gettoni (MATRICE di gioco PIENA) che la partita è finita alla pari.

2) **TriClient** :Il client dovrà supportare alcune opzioni in fase di esecuzione, la prima è il nome del giocatore, quindi la riga di esecuzione sarà:

./TriClient nomeUtente

Una volta lanciato il client, rimarrà in attesa che venga “trovato il secondo giocatore” dopo di che il gioco potrà iniziare (è opportuno che al client venga notificato la ricerca di un giocatore per proseguire). Il client si occupa di stampare a ogni turno “la matrice” di gioco aggiornata, e di chiedere al giocatore su quale colonna intende inserire il proprio gettone. Si noti che i client NON imbrogliano, ma di fatto devono segnalare al giocatore se la POSIZIONE prescelta non è utilizzabile perché già piena. È discrezione del candidato, decidere eventuali “penalità” se si inserisce il gettone in una posizione già occupata (ad esempio si potrebbe passare il turno all’altro giocatore, se si posiziona in un posto già occupato). La pressione di CTRL-C sul client andrà gestita, di fatto verrà considerato che il giocatore perde “per abbandono” della partita; quindi, il client dovrà prima di terminare notificare la chiusura anticipata della partita al server (che a sua volta notificherà al secondo giocatore la vittoria per abbandono dell’altro giocatore).

FUNZIONAMENTO DEL CODICE:

Iniziamo l’analisi del codice da TriServer composto da due parti. La prima è deputata alla gestione della partita nel caso in cui fossero presenti due giocatori mentre la seconda al ricevimento di un flag specifico dal client va a duplicare il server tramite una fork e attraverso una exec va ad eseguire il codice del client allo scopo di creare un giocatore automatico. Sono presenti svariati controlli in quanto il server si occupa di controllare ad ogni turno l’eventuale vittoria di uno dei due giocatori o del bot.

TriClient invece si divide in svariate parti dal momento che nel caso del funzionamento base abbiamo un controllo sull’identità del giocatore (1 o 2) basato sul valore dei semafori. L’altra parte, ha un compito più particolare dal momento che si occupa di alzare un flag nella memoria condivisa quando nella riga di comando viene inserito un asterisco il quale porterà alla creazione di un bot.È stato utilizzato un sistema di tre semafori dedicati rispettivamente al Server , Client2 e Client1 che vanno a sbloccare e bloccare i processi gestendo così la partita.Fondamentale è stato l’utilizzo della memoria condivisa la quale permette di condividere la matrice che fa da sfondo al gioco stesso assieme ad una serie di parametri di controllo e non solo.

SCELTE PROGETTUALI

Dato che nel testo erano presenti dei punti ambigui sono state prese delle scelte progettuali tra cui:

1 Nel momento in cui scade il timeout per un client il server va a interrompere la partita assegnando la vittoria all’altro giocatore.

2 Quando uno dei due giocatori vince il gioco mette a disposizione due scelte :

- Proseguire ugualmente la partita nonostante la sconfitta
- L’abbandono della partita che verrà notificato all’altro giocatore con annesso messaggio di vittoria.