

# Relazione di Laboratorio di Sistemi Operativi

## Partecipanti al progetto

Luca Bianchi, Matricola: 114150

## Progetto

Il nome del progetto è CTris, in quanto come suggerisce il nome è un'implementazione del gioco del Tris (anche detto Tic Tac Toe) in linguaggio C.

## Specifiche

Il progetto CTris riguarda la realizzazione dell'omonimo gioco realizzato per mezzo di una comunicazione Client/Server.

Più precisamente è costituito da un client (che sarebbe il giocatore che richiede di iniziare una partita) ed un server che invece resta in ascolto di eventuali connessioni da parte di un client.

## Server

Il server (presente nel file `ctris_server.c`) una volta eseguito è un processo che resta in attesa sulla porta 8080 (di default).

La particolarità del server in questione è che è un server "concorrente" il che vuol dire che una volta che un client richiede di potersi connettere, il server con la system call `fork()` andrà a generare un processo figlio, e sarà poi lui stesso a gestire il gioco effettivo comunicando con il client (ed avviene il tutto in modo automatico).

Il server così facendo potrà tornare con il processo padre in attesa di una nuova connessione da parte di un client.

Ovviamente la system call più importante in tutto questo progetto è la `socket()`, per mezzo della quale possiamo inizializzare un nuovo socket dove potrà mettersi in ascolto il server.

La modalità di connessione utilizzata per la comunicazione è `SOCK_STREAM`, vale a dire una comunicazione TCP a tutti gli effetti (che si differenzia per questo da una UDP).

Nel server, la prossima mossa (valida) da parte della computer viene scelta per mezzo di un numero casuale per semplicità.

## Esecuzione

Il server può volendo essere eseguito su un'altra porta, che viene passata come argomento all'esecuzione dell'eseguibile, ad esempio scrivendo a linea di comando

```
./ctris_server 4444.
```

L'importante ovviamente è che client e server si connettano sulla stessa porta (quella dove il server è in ascolto che di default è 8080), altrimenti i due rispettivi processi non potranno comunicare.

## Client

Il client (presente invece del file `ctris_client`) è il programma che dovrà essere eseguito dal giocatore che vuole giocare contro il computer.

Il client a sua volta dovrà utilizzare la system call `socket()`, dato che dovrà inizializzare un socket per poter provare a connettersi ad un server, di cui deve essere fornito l'indirizzo (che di default è 127.0.0.1, quello di localhost) e la relativa porta dove il servizio è in ascolto (che di default è 8080).

Dopo una fase di "inizio comunicazione", quando comincia la partita vera e propria, il server si limita a mandare un array di caratteri contenente la board, quindi è il client ad avere la responsabilità di formattare la stampa della board del gioco.

## Esecuzione

Come già detto, anche il client può essere eseguito con parametri diversi.

Infatti per poter eseguire un client che si vuole poter connettere ad un servizio presente su un altro host e su una certa porta, possiamo eseguire ad esempio

```
./ctris_client 192.168.0.2 4444
```

## Gioco

Nel gioco, la comunicazione avviene nel seguente modo:

- Il server invia al client direttamente l'intera board sotto forma di un array di `char`, mentre il client una volta ricevuta si preoccuperà di stamparla con un certo formato.
- Il client invece si limita ad inviare al server un `char` contenente la mossa scelta.

Per rendere il server un po' più snello alcuni controlli (come ad esempio se la mossa scelta dal client è valida o meno) vengono fatte direttamente dal client.

Per comodità, il giocatore comincerà sempre per primo ed ha assegnato come simbolo "X", mentre il computer utilizzerà il simbolo "O".

Inoltre per semplicità una volta terminata la partita, mentre il server è sempre comunque in attesa di un nuovo client, il client che ha terminato la partita termina il relativo processo completamente (così come il processo figlio del server che se ne occupava).

Quindi per iniziare una nuova partita, il processo client dovrà essere eseguito nuovamente.

Infine la particolarità della board è che venendo rappresentata come un array di 10 `char`, il primo carattere (cioè `board[0]`) viene sempre utilizzato per comunicare lo “stato” della partita, cioè:

- V: Vittoria
- S: Sconfitta
- P: pareggio
- O: se sta ancora procedendo.

Perciò mentre è il server a controllare se la partita è finita o meno (e nel caso in cui sia terminata imposta di conseguenza il primo carattere della board), il client si limita a controllare il risultato inviato dal server e comunicare il risultato a console se necessario.