

ELABORATO FINALE SYSTEM CALL

Corato Francesco [VR489009]
Sinico Enrico [VR488626]

FUNZIONAMENTO DEL PROGRAMMA

Una volta compilato verranno creati due file oggetto, **TriServer** e **TriClient**, nella stessa cartella (l'utente deve avere i permessi di lettura, scrittura, e esecuzione nella cartella).

L'esecuzione del programma è composta dell'avvio di un TriServer e due TriClient.

Il TriServer deve essere avviato da linea di comando con i seguenti parametri:

- **Secondi di time-out** per fare una mossa (se impostato a 0 non c'è limite di tempo);
- **Simbolo giocatore 1** (un solo carattere, nel caso venga messa una stringa verrà preso solo il primo);
- **Simbolo giocatore 2** (analogo a giocatore 1).

Il TriServer rimarrà in attesa dell'avvio dei due TriClient.

I TriClient possono essere avviati in linea di comando con i seguenti parametri:

- **Nome giocatore** (i due giocatori possono avere lo stesso nome);
- (facoltativo) **Modalità di esecuzione:**
 - **' '** (vuoto) su Giocatore 1 se vuole giocare contro una persona;
 - **'*'** su Giocatore 1 se vuole giocare contro un bot: in questo caso il Giocatore 2 verrà creato e controllato dal Server;
 - **'H'** su Giocatore 1 se vuole essere controllato da un bot e giocare contro un bot;
 - **'*'** o **'H'** su Giocatore 2 non funzionano in quanto esiste già un Giocatore 1, non può quindi giocare contro un **bot** (giocherà contro Giocatore 1).

Il Server utilizza un semaforo per capire quando entrambi i Client sono connessi, i Client a loro volta usano un semaforo a testa per dichiarare al Server quando sono connessi.

Se esistono già due Client e se ne apre un terzo, questo manderà un avviso e non si aprirà.

Una volta connessi entrambi i Client il Server avvierà la partita (viene avviata tramite l'utilizzo di un semaforo per cui i Client sono in attesa).

I Client possono essere avviati solo dopo il Server, altrimenti stamperanno un messaggio di errore e termineranno (è il server che inizializza tutti i semafori).

I Client possono fare una mossa solo nel proprio turno (a meno che non sia qualcosa di speciale come la resa), la prima mossa spetta a Giocatore 1. Questo viene fatto tramite due semafori (uno per Client).

Una mossa normale consiste nel inserire un numero [1, 9] corrispondente a una casella della tabella (campo di gioco) visualizzata a schermo.

Inserire un carattere non valido comporta il salto del proprio turno come penalità.

Se il time-out (primo parametro inserito all'avvio del TriServer) per compiere una mossa si esaurisce, il giocatore perde automaticamente la partita.

Per arrendersi, il Client non deve far altro che premere **Ctrl-C** (mandando un segnale **SIGINT**).

Chiudere il terminale premendo la **X** di chiusura della finestra verrà contato come resa.

Se viene premuto **Ctrl-C** nel Server, verrà visualizzato un messaggio che avvisa che premendolo una seconda volta chiuderà il Server, terminando la partita.

SCELTE PROGETTUALI

- Compiere una mossa non valida fa saltare il turno.
- Quando una partita finisce bisogna riavviare tutto il programma per cominciarne un'altra.
- Se non si compie una mossa entro il limite di time-out si perde automaticamente.
- Per quanto riguarda la rimozione degli **IPCS**, nel caso l'utente chiuda il programma mandando un segnale **SIGKILL** al server, gli **IPCS** rimarranno nella memoria del calcolatore fino alla riapertura del programma (che li rimuove e poi li ricrea) o allo spegnimento del calcolatore. Se invece viene mandato ai Client si bloccano in quanto non è previsto che sappiano della disconnessione l'uno dell'altro. Nel caso in cui il time-out sia attivo, al termine dei secondi impostati verrà comunque terminata la partita.
- Abbiamo implementato la possibilità di far controllare il Client da un **bot** immettendo come secondo parametro '**H**', questo permette di creare una partita **bot vs bot** mettendo '**H**' nel Client 1.
- In alcuni casi, ad esempio quando scade il tempo di una mossa contro un bot, il programma si chiude correttamente ma non viene visualizzato il prompt nel terminale del server, tuttavia se si prova a lanciare un comando funziona, è solo un bug grafico legato al terminale (con il terminale **gnome-terminal** ha questo bug, mentre il terminale **kitty** visualizza correttamente il prompt).