

RELAZIONE PROGETTO DI SISTEMI OPERATIVI 2022/2023

Kovacs Alexandru Ioan VR471323

June 15, 2023

0.1 Struttura del progetto

File:

- Inc.
 - "all-lib.h"
 - "fifo.h"
 - "my-own-library-final.h"
- Src.
 - server.c
 - client.c
 - fifo.c
 - my-own-library.c
- No-folder.
 - Makefile
 - F4server
 - F4client

Compilazione: Per compilare:

make

Esecuzione: Per eseguire il programma:

Facciamo partire prima il server

Sul primo terminale ./F4server dim1 dim2 token1 token2

Per far connettere due giocatori apriamo due terminali diversi ed eseguiamo il seguente codice:

```
./F4client nomegiocatore  
./F4client nomegiocatore
```

Per far connettere un giocatore contro un bot eseguiamo il seguente codice:

```
./F4client nomegiocatore \*
```

Fine: Per cancellare gli eseguibili creati dal make file basta digitare il comando:

```
make clean
```

0.2 Scelte progettuali

- Inizialmente ho sviluppato il gioco di forza4 in c scrivendo una libreria di funzioni che potesse tornarmi utile per il progetto
- Per la connessione dei due client al server ho utilizzato una FIFO
- Per la comunicazione server to client ho utilizzato una MsgQueue
- Ho impostato una dimensione massima della matrice prestabilita (20x20)
- Il server inizializza la partita creando la memoria condivisa e inizializzando la matrice solo per la dimensione indicata quando esso viene eseguito
- Siccome le funzioni che ho scritto per forza4 lavorano su una matrice e la memoria condivisa ha un puntatore ad essa ho preferito lavorare su una copia locale di essa per poi copiarla all'interno della memoria condivisa
- I giocatori che sono in attesa di connessione vengono sbloccati da un segnale inviato dal server
- Se un terzo giocatore provasse a connettersi durante una partita, verrà informato che esiste già una partita in corso e che è piena
- Il segnale in questione per quanto riguarda due giocatori è SIGUSR1 mentre per l'opzione del bot viene invitato SIGTERM (SIGUSR1 non lo rileva)
- Il server fa da arbitro e controlla dopo l'inserimento di ogni giocatore la vittoria
- Se qualcuno ha vinto il server si occupa di:
 - Stampare a video il numero del giocatore vincente
 - Informare i giocatori che un giocatore ha vinto quindi la partita è terminata
 - Rimuovere le IPCS presenti
- Se nessuno ha vinto invece viene ceduto il turno al prossimo giocatore, per i turni ho utilizzato dei semafori ed una variabile turno
- Per il doppio Ctrl+C del server ho utilizzato un contatore

- Per il Ctrl+C del client ho deciso che alla singola pressione il giocatore perde per abbandono, il server viene informato e si occupa di:
 - Sconnettere l'altro giocatore
 - Eliminare le IPCS
- Il client svolge queste principali attività:
 - Esegue la giocata
 - Controlla il pareggio ed in caso affermativo informa il server
 - Informa il server se uno dei due giocatori ha abbandonato
 - Informa il server se uno dei due giocatori non ha eseguito la mossa entro lo scadere del tempo, in caso si verificasse, disconnette il giocatore ed informa il server
- Time-Out: se un giocatore non esegue la sua mossa entro un tempo limite impostato (5secondi) verrà automaticamente disconnesso e perderà così per abbandono
- variante Time-Out: se un giocatore esegue la mossa dopo lo scadere del tempo verrà disconnesso e perderà in automatico la partita, questo comporta una vittoria a tavolino dell'altro giocatore