

UNIVERSITÀ DEGLI STUDI DI VERONA

Elaborato System Call

Sistemi Operativi - Laboratorio

Corso di Laurea in Informatica

Jonathan Fin (VR471297) e Roberto Trinchini (VR472851)

19/06/2023

Forza 4

Indice

Elenco dei file	3
Descrizione dell'avvio del programma	3
Funzionamento dei processi	4
Gestione dei segnali	6
Gestione delle zone di memoria	6
Schema generale	7

Elenco dei file

- **server.c** → programma che svolge le istruzioni descritte relative al processo Server
- **client.c** → programma che svolge le istruzioni descritte relative al processo Client
- **bot.c** → programma usato per calcolare una mossa in maniera automatica
- **errExit.h** → comprende una sola funzione (`errExit`), usata per far terminare un processo stampando il corretto messaggio di errore
- **semaphore.h** → comprende funzioni di supporto per la gestione dei semafori
- **shared_memory.h** → comprende funzioni di supporto per la gestione delle zone di memoria condivisa, e la struttura dati della memoria principale
- **playground.h** → comprende funzioni e variabili usate per gestire il campo da gioco (creazione della zona di memoria condivisa, mosse, controlli, ...)
- **terminal.h** → comprende funzioni di supporto per impostare il terminale (per far in modo che non stampi a video lo standard input)
- **graphic.h** → comprende funzioni e variabili per la gestione della parte grafica (frontend)

*tutti i file .h comprendono anche i relativi file .c

Descrizione dell'avvio del programma

- **Server** → `./F4Server <n° righe> <n° colonne> <simbolo giocatore 1> <simbolo giocatore 2> <Key>`
 - Le righe e colonne devono avere entrambe valore minimo 5 e massimo 20
 - I simboli dei giocatori devono essere di 1 carattere
 - Key indica la key della zona di memoria condivisa principale, deve essere maggiore di 0 (e minore di 1 000 000)
- **Client** → `./F4Client <Key> <Nome> <Opzione Bot> <Bot Log>`
 - Key è la stessa del Server, usata per connettersi alla stessa zona di memoria
 - Nome indica il nome del giocatore
 - Opzione Bot indica se il Client deve essere controllato dal Server; si indica con `*` se si vuole usare il Bot, altrimenti non si scrive niente
 - Bot Log è un'opzione (si può anche non scrivere), indica se il Bot deve scrivere sulla console (del Server) le mosse che farà (se non sono casuali). Si indica con `on` oppure `off`. Se non si scrive niente viene impostato di default a `off`

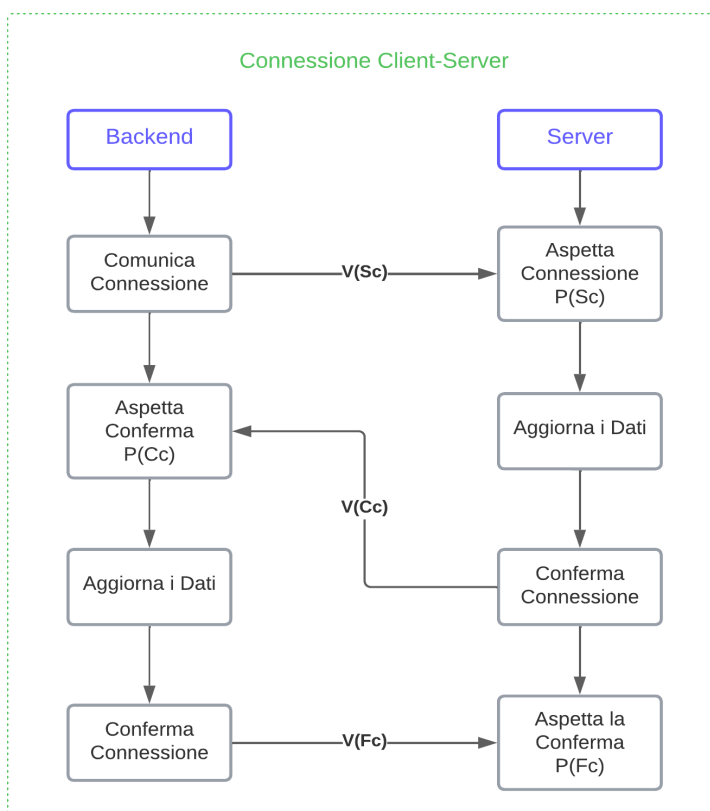
Funzionamento dei processi

Server:

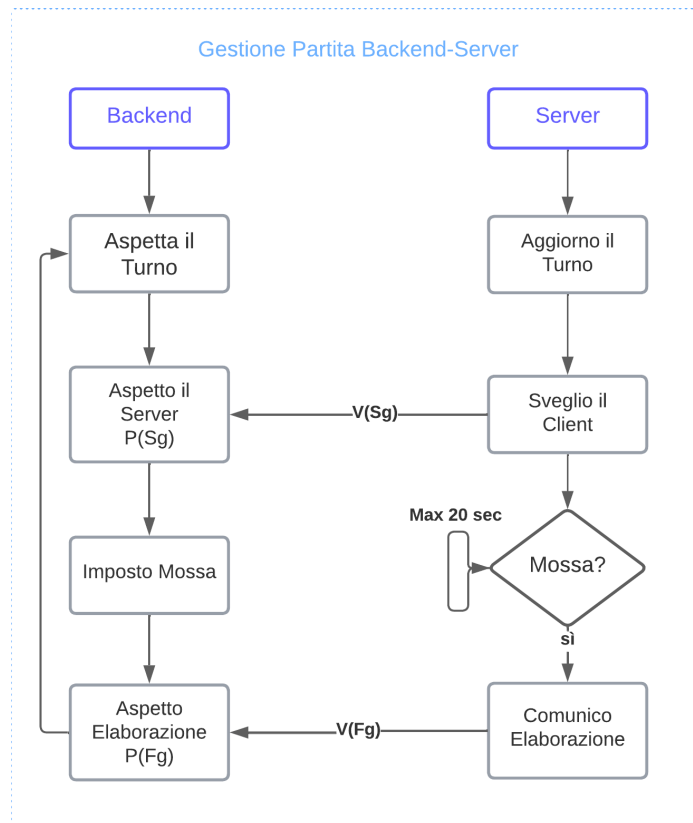
- Controlla gli argomenti
- Crea il campo da gioco (zona di memoria condivisa) e i semafori per la sincronizzazione
- Aspetta la connessione di entrambi i Client e imposta i loro valori quando si connettono
- Gestisce i turni di gioco
 - Ogni client ha massimo n (20) secondi prima del timeout
 - Se scade il tempo, la partita finisce, con la vittoria dell'altro giocatore
 - Se il client ha inserito l'opzione bot:
 - Il Server si divide in 2 e il figlio esegue il file eseguibile del bot
 - Il processo Bot restituisce la mossa
 - Ogni turno il Server controlla se il giocatore che ha effettuato la mossa ha vinto, oppure se il campo da gioco è pieno (pareggio)

Client:

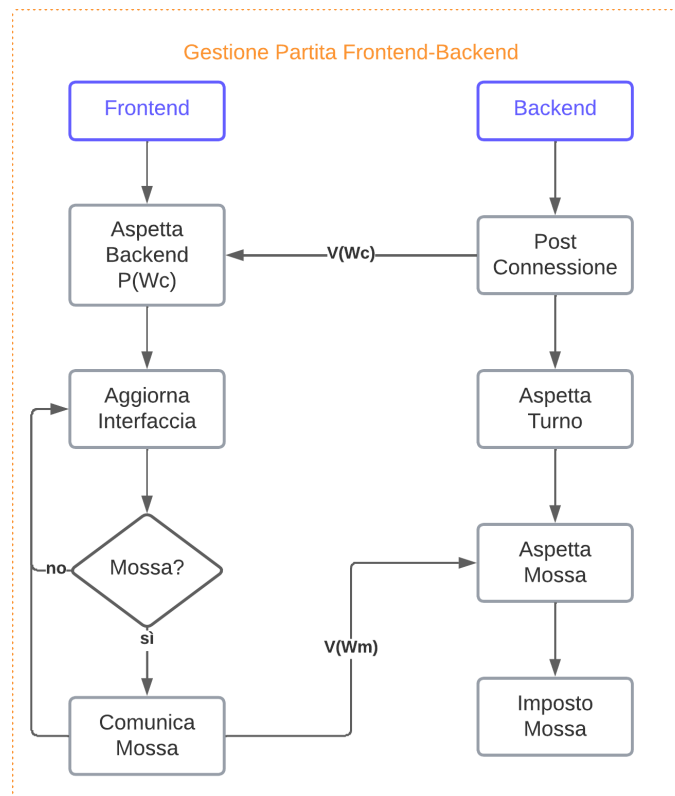
- Controlla gli argomenti
- Si connette alla zona di memoria principale
- Si divide in 2 processi:
 - Backend / figlio, il controllore che interagisce con il server
 - Frontend / padre, che stampa il campo da gioco
- Il frontend deve aspettare che il backend si connetta al Server
- Il backend si connette con il Server (3-way handshake)



- Dopo aspetta il server per iniziare il gioco



- Una volta iniziato il gioco, l'utente può scegliere in quale colonna inserire il proprio gettone (se è il suo turno) e il Client invia la mossa al Server



Gestione dei segnali

Server:

- **SIGUSR2** → Segnale inviato da un Client; indica che un Client si è disconnesso
 - Se avviene prima che inizi la partita, il Server aspetta un altro Client
 - Se avviene a partita iniziata, la partita termina e l'altro Client vince a tavolino
- **SIGINT** → Controlla quante volte è stato premuto `Ctrl + C` in un certo periodo di tempo
 - Se si preme 2 volte entro n (3) secondi, il processo Server termina e i Client verranno disconnessi
- **SIGALRM** → Questo segnale viene mandato dall'handler del **SIGINT**; resetta il contatore del `Ctrl + C`

Client:

- **SIGINT** e **SIGHUP** → Indica che un Client vuole disconnettersi
 - Invia un segnale **SIGUSR2** al Server e termina il processo
 - Nel processo figlio termina il processo direttamente, senza inviare segnali
- **SIGUSR1** → Segnale inviato dal Server; indica che è finita la partita
 - Termina il processo
- **SIGUSR2** → Segnale inviato dal Server; indica che l'altro Client si è disconnesso oppure che il Server è terminato
 - Termina il processo

Gestione delle zone di memoria

- **Data** → zona di memoria principale; contiene:
 - Dati dei giocatori
 - PID del Server
 - Colonna nella quale è stata effettuata la mossa
 - Semafori
 - Zona di memoria del campo da gioco
- **Playground** → zona di memoria del campo da gioco; contiene:
 - Dimensione del campo
 - Zone di memoria delle colonne libere e della matrice del campo da gioco
- **GraphicData** → zona di memoria per la grafica; contiene:
 - Id del giocatore (0 o 1)
 - Valore booleano che indica se è il turno del giocatore
 - Indice della colonna attualmente selezionata dal giocatore

Schema generale

