

Relazione Sistemi Operativi

Vincenzo Sanzone

July 7, 2023

Contents

1	Server	2
1.1	Introduzione generale	2
1.2	IPC	2
1.3	Segnali	2
1.4	System Call	3
2	Client	3
2.1	Introduzione generale	3
2.2	Segnali	3
2.3	Client automatico	3

1 Server

1.1 Introduzione generale

Il server viene eseguito attraverso il comando `"/F4server n m {simbolo-1} {simbolo-2}"` dove n indica le righe, mentre m le colonne. Dopo essersi assicurato della correttezza, il server inizierà il gioco, permettendo ai client di fare le loro mosse. Ogni mossa verrà controllata dal server, al fine, di verificare l'eventuale fine della partita.

1.2 IPC

Al fine di un corretto funzionamento del programma, il server deve comunicare alcune informazioni, o sincronizzarsi con il client, dunque vengono usate le IPC, nello specifico:

- **Memoria condivisa:** usata per la condivisione del tabellone del gioco, e di altre informazioni, come, ad esempio le coordinate del gettone giocato e il simbolo di chi lo gioca;
- **Semafori:** usati per la sincronizzazione server-client, ad esempio quando si è in attesa della mossa/turno;
- **Coda dei messaggi:** usata per inviare il messaggio di fine partita, sia essa per parità, vittoria, sconfitta, abbandono o time-out.

Tutte le IPC vengono create attraverso una chiave casuale generata da `IPC_PRIVATE`, e l'identificatore viene condiviso attraverso il file di testo `key.txt`.
Con la terminazione del programma, il server andrà a rilasciare tutte le IPC usate.

1.3 Segnali

Il server gestisce in modo autonomo i seguenti segnali:

- **SIGHUP:** avviso i client e rimuovo le risorse;
- **SIGINT:** aspetto due volte questo segnale prima di agire come SIGHUP. Il counter viene resettato ogni x secondi, o al ricevimento di un altro segnale. x è un numero che si può scegliere prima della compilazione;
- **SIGALRM:** uno dei client non ha risposto in tempo, dunque li avviso e rimuovo le risorse;
- **SIGUSR1:** un client ha abbandonato, quindi avviso l'altro, se esiste, e rimuovo le risorse;
- **SIGUSR2:** il client è stato avviato in modalità gioco automatico, quindi creo il secondo "client".

1.4 System Call

Durante la sua vita, il server potrebbe usare le seguenti chiamate a sistema operativo:

- **Read:** viene usata per leggere gli interi dal file *key.txt*;
- **Write:** viene usata per scrivere gli interi nel file *key.txt*;
- **Fork:** usata per creare un processo figlio, in caso di client automatico;
- **Exec:** usata dopo la fork per eseguire "F4auto_play";
- **Wait:** necessaria per aspettare la terminazione del processo figlio;
- **Gestione:** segnali, IPC, file.

2 Client

2.1 Introduzione generale

Il client viene eseguito attraverso il comando `"/F4client {nome} *(opzionale)"`. Dopo essersi assicurato della correttezza, il client si conatterà al server, aspettando il secondo giocatore o il suo turno.

Le IPC, così come la maggior parte delle, System Call usate corrispondono a quelle del server.

2.2 Segnali

Il client gestisce in modo autonomo i seguenti segnali:

- **SIGHUP/SIGINT:** avviso il server del mio abbandono, e scollego le risorse condivise;
- **SIGTERM:** il server è stato terminato, quindi scollego le risorse;
- **SIGUSR1:** la partita è terminata, quindi stampo l'esito e scollego le risorse.

2.3 Client automatico

L'uso di `*` dalla riga di comando comporta che vengano passati tutti i file all'interno della cartella. Per assicurarsi che sia stato lanciato correttamente, quindi, il client aprirà la cartella, e conterà tutti i file regolari, assicurandosi che corrispondano ai parametri passati. Se i parametri passati saranno di più o di meno rispetto a quelli previsti ($argc - file_regolari = 2$), allora terminerà. Il client automatico funziona in grandi linee come se fosse un client normale.