

Progetto NIM - Sistemi Operativi

Università degli Studi di Udine Corso di Laurea triennale in informatica Progetto NIM - Laboratorio di Sistemi Operativi.

Prof. Ivan Scagnetto & Dott. Nicola Gigante.

A.A. 2019-2020 - Primo Semestre.

Studenti: Alberto Morini (141986), Davide Vezzaro (142786)

14/12/2019

Suddivisione del lavoro.

Progettazione:

Dopo aver studiato il funzionamento del gioco matematico NIM e comprese le sue regole, abbiamo iniziato a definire i compiti e azioni che Server e Client avrebbero dovuto eseguire.

Al fine di favorire una buona cooperazione durante lo sviluppo del codice abbiamo creato una "Repository" su GitHub.

Successivamente abbiamo progettato la comunicazione socket client-server tramite l'invio delle struct "partitaGioco" con lo scopo di rappresentare lo stato della partita (turno del giocatore, pedine presenti ecc.) e la struct "scelta" contenente la mossa del client (pila selezionata, numero di pedine da rimuovere)

Implementazione:

In primis abbiamo definito l'header dei due programmi e inoltre un file di intestazione comune (common.h) contenente la dichiarazione di struct e costanti necessarie sia al server che al client.

Successivamente abbiamo iniziato a sviluppare il progetto implementando il codice, aiutandoci e consultandoci ad ogni esigenza.

Mentre per la compilazione dei programmi abbiamo creato un "makefile" principale il quale genera i rispettivi "nimserver" e "nimclient".

Test:

A termine del progetto è stato verificato il corretto funzionamento del gioco, controllando anche le casistiche di errore da parte dell'utente (scelta/azione non valida) o di assenza di connessione dovuta alla terminazione di un processo. In fine ci siamo assicurati che il server chiudesse gentilmente le connessioni con i rispettivi clients una volta raggiunto il termine partita.

Tecnologie utilizzate:

GitHub per una repository comune. Visual Studio Code per lo sviluppo, versione 1.41.1

MacOS Catalina(10.15.2) con versione clang Apple: 11.0.0 Ubuntu eoan (19.10) clang version 9.0.0-2

2 14/12/2019

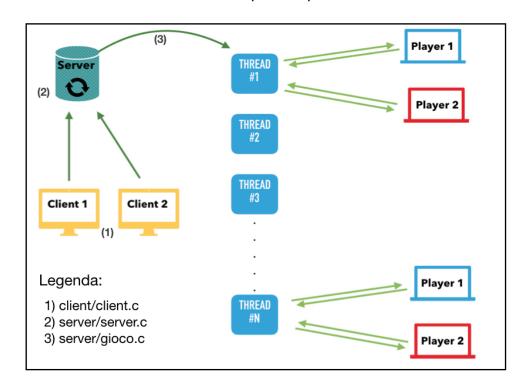
Struttura del progetto

Server:

- Riceve la connessione dai client e successivamente avvia la partita su un thread dedicato a due giocatori.
- Inizializza il gioco generando casualmente il numero di pedine che compongono le due pile, le comunica ai client e attende la mossa dal primo giocatore (il primo che si è connesso).
- Ricevuta l'azione di un giocatore, convalida la correttezza della mossa e nel caso positivo aggiorna lo stato partita, altrimenti segnala l'errore e richiede nuovamente l'azione al client.
- Inoltre verifica la connessione con i client, se un giocatore dovesse chiudere la propria comunicazione la vittoria sarà assegnata all'avversario.
- Nel caso di vittoria ha il compito di comunicare ai giocatori il risultato, per poi terminare l'esecuzione del thread.
 Mentre l'esecuzione del server rimane sempre in ascolto di nuove connessioni.

Client:

- Avvia la connessione con il server e attende la creazione della partita, per poi stampare a video lo stato del gioco.
- Chiede all'utente la mossa che desidera eseguire, quindi la Pila selezionata e il numero di pedine da rimuovere.
- Riceve dopo ogni mossa (sia dell'avversario che sua) l'attuale stato della partita.
- Comunica all'utente il vincitore della partita quando essa è terminata.



3 14/12/2019