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)

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 le azioni che Server e Client avrebbero dovuto eseguire.
Successivamente abbiamo creato una “Repository" su GitHub al fine di poter lavorare sulla versione più recente del progetto senza tante complicazioni.

Abbiamo deciso di effettuare la comunicazione client-server tramite struct, in modo da semplificare lo scambio di dati, comunicando in un’unica volta tutte le informazioni della partita o dell’azione effettuata.

Implementazione:

In primis abbiamo definito le interfacce/librerie che i rispettivi file avrebbero incluso, quindi le funzioni e procedure che client e server necessitano per la loro esecuzione.
Successivamente abbiamo iniziato a sviluppare il progetto implementando il codice, aiutandoci e consultandoci ad ogni esigenza.

Test:

A termine del progetto, abbiamo avviato la fase di test; eseguendo il programma più volte, verificando il corretto funzionamento e i possibili errori che l’utente potrebbe commettere.
Verificando anche le casistiche di assenza di connessione nel momento in cui un client dovesse terminare il proprio processo.
Assicurandoci inoltre che il server chiuda gentilmente la connessione con i rispettivi clients.

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

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).
- Dopo aver ricevuto 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 del 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 vuole fare, 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 e aggiorna il video comunicando all'utente il caso di vittoria o sconfitta.