SPECIFICHE PROGETTO - RETI INFORMATICHE cod. 545II, 9CFU
A.A. 2024/2025

Si richiede di implementare un'applicazione distribuita che implementi un **sistema di gaming distribuito** (gioco client-server).

Descrizione

Realizzare un gioco **Trivia Quiz Multiplayer.** Il gioco prevede domande e risposte. Il server pone domande a tutti i client connessi. I giocatori inviano le loro risposte al server, che le valuta e aggiorna il punteggio, mantenendo una classifica di tutti i giocatori.

Ci devono essere almeno due temi per le domande:
es: (1) Curiosità sulla tecnologia, (2) Cultura Generale.

Ci devono essere 5 domande per tema:
es: (1) Chi ha fondato l'azienda Microsoft? Risposta: Bill Gates e Paul Allen;
(1) Quale azienda ha introdotto il primo personal computer commerciale nel 1981?
Risposta: IBM;
(2) Quale oceano è il più grande del mondo? Risposta: Oceano Pacifico
(2) Quale è il fiume più lungo del mondo? Risposta: Nilo

Ogni client deve decidere a quale quiz vuole partecipare (può partecipare a un solo quiz per volta), e ogni tema avrà la sua classifica.

Viene assegnato 1 punto per risposta corretta, 0 per risposta sbagliata.
Si può partecipare solo una volta ad ogni singolo quiz.

Per l'implementazione, scegliere i protocolli di comunicazione, le strutture dei pacchetti, le modalità di scambio dei dati (text o binary), e la tipologia di server (iterativo, concorrente, o basato su I/O multiplexing) dipendentemente da aspetti che si ritengono congrui all'applicazione descritta. Le considerazioni fatte per prendere tutte queste decisioni devono essere spiegate in una documentazione da consegnare assieme ai sorgenti.

Documentazione

L'applicazione distribuita deve implementare quanto descritto in *"Descrizione"*. Le scelte progettuali devono essere spiegate in una relazione di non più di 2 pagine, font Arial, dimensione 11 pt, usando, all'occorrenza, anche figure e schemi intuitivi.
Nella relazione devono essere messi in luce, in modo critico, potenziali pregi e difetti delle scelte fatte. Documentazioni contenenti solo le scelte fatte, senza un'analisi critica, non saranno considerate sufficienti: dovranno essere integrate in sede di orale. Se la documentazione consegnata contiene più di due pagine, la valutazione del progetto avverrà considerando solo le prime due.

Schema funzionamento applicazione

Il **server** è mandato in esecuzione sull'host come segue:

./server

Appena mandato in esecuzione, il server mostra a video (i) la lista dei temi del quiz e (ii) il numero e il nome dei client connessi.
Dopo la ricezione di un messaggio dalla rete da parte dei client, il server mostra a video la lista aggiornata di giocatori, e include una sezione "Punteggio" con i giocatori ordinati per punteggio decrescente, e due sezioni per tener traccia dei giocatori che hanno completato un quiz.

```
Trivia Quiz
++++++++++++++++++++++++++++++
Temi:
1 - Curiosità sulla tecnologia
2 - Cultura Generale
++++++++++++++++++++++++++++++

Partecipanti (2)
- Pippo
- Pluto

Punteggio tema 1
- Pippo 3
- Pluto 1

Punteggio tema 2
- Pippo 1

Quiz Tema 1 completato
- Pippo

Quiz Tema 2 completato
------
```

Va gestita in maniera corretta, un'eventuale disconnessione/chiusura di socket nel server. I Client dovranno essere avvisati che il server si è disconnesso (o il socket è stato chiuso), e che quindi il quiz è finito.
Le domande e le risposte dei due quiz, sono salvate in due file, ai quali il server deve accedere per fornire le domande al client e controllare la correttezza delle risposte.

Un **client** è mandato in esecuzione sull'host come segue:

./client <porta>

Dove <porta> è la porta che il server usa per esporre il servizio del gioco. All'avvio, il client mostra un menù testuale con la possibilità di cominciare un quiz, o di uscire. La scelta dell'opzione 1, manderà al server la richiesta di ricevere informazioni sui quiz attualmente disponibili, e di fatto stabilirà la connessione con il server. L'opzione 2 terminerà il client.

Se il client sceglie l'opzione 1, il server gli chiederà un nickname, che deve essere unico tra tutti i partecipanti al quiz. Il server dovrà gestire questi controlli, e dare la possibilità al client di cambiare nickname finché non ne trova uno libero.

Una volta scelto il nickname, il client dovrà scegliere a quale quiz partecipare.
Una volta scelto, l'informazione verrà inviata al server, che gli fornirà la prima domanda del quiz, e rimarrà in attesa che il client risponda.
Il server poi, ricevuta la risposta, ne controllerà l'esattezza, e risponderà al client con "Risposta corretta", o "Risposta errata", aggiornando coerentemente il punteggio del client, per ogni tema.

```
Trivia Quiz
++++++++++++++++++++++++++++++
Menù:
1 - Comincia una sessione di Trivia
2 - Esci
++++++++++++++++++++++++++++++
La tua scelta:
```

```
Trivia Quiz
++++++++++++++++++++++++++++++
Scegli un nickname (deve essere univoco):
```

```
Quiz disponibili
++++++++++++++++++++++++++++++
1 - Curiosità sulla tecnologia
2 - Cultura Generale
++++++++++++++++++++++++++++++
La tua scelta:
```

```
Quiz - Curiosità sulla tecnologia
++++++++++++++++++++++++++++++
Quale azienda ha introdotto il primo personal computer commerciale nel 1981 ?

Risposta:
```

Durante la sessione di quiz, il client può decidere di voler conoscere i punteggi degli altri partecipanti al quiz. Può farlo con il comando "**show score**". Il server risponderà con i punteggi di tutti i client per ogni tema.

Durante la sessione di quiz, il client può decidere di terminare digitando il comando "**endquiz**".
A quel punto, il server che riceverà il comando "endquiz", chiuderà la comunicazione con il client, eliminando punteggi e nickname.
Al client verrà visualizzato nuovamente il menù di avvio.

Per semplicità, server e client sono eseguiti sullo stesso host. Per questo, il comando di avvio del client non necessita di un parametro per specificare l'IP del server (che sarà 127.0.0.1).

REQUISITI

- I dati sono scambiati tramite socket. Se non si definisce un formato di messaggio, prima di ogni scambio, il ricevente deve essere informato su quanti byte leggere dal socket.
- Per gestire le varie informazioni, è possibile usare strutture dati a piacere. Gli aspetti che non sono dettagliatamente specificati in questo documento possono essere implementati liberamente.
- Il codice deve essere indentato e commentato in ogni sua parte: significato delle variabili, processazioni, ecc. I commenti possono essere evitati nelle parti banali del codice.
- Va prodotta una documentazione di 2 pagine (font Arial, size 11 pt) in cui giustificare, in maniera critica, le scelte fatte, e i formati di strutture dati e messaggi (ove introdotti).

CONSEGNA

- Il progetto deve essere caricato attraverso il **modulo google**
  - https://forms.gle/J7KymCTekD4gAQ2Z6), **entro 5 giorni** dal giorno dell'esame.

  - Esempio: Appello 8/01, il progetto va consegnato entro le 23:59 del 03/01. Il form non accetterà più sottomissioni dopo quella data/ora.

- Il progetto va consegnato nello stesso appello in cui si ha intenzione di sostenere l'orale.

VALUTAZIONE

- Il progetto è visionato e valutato prima dello svolgimento dell'esame, eseguendolo su sistema operativo Ubuntu 24, distribuzione usata durante il corso. Testare sempre il codice (compilare ed eseguire) su una macchina Ubuntu 24 prima della consegna.

- Durante l'esame, può essere richiesta l'esecuzione del programma, modifiche, e saranno fatte domande sia sul codice che sulle scelte fatte.

- Verranno fatte verifiche di eventuale plagio, su tutti i progetti consegnati.

- La valutazione del progetto prevede le seguenti fasi:

  - Compilazione del codice:
    - Il client e il server vanno compilati con opzione -Wall che mostra i vari warning. **Non vi dovranno essere warning o errori**. L'opzione -Wall va abilitata anche durante lo sviluppo del progetto, interpretando i messaggi forniti dal compilatore. **Se il progetto non compila, non è valutabile e non si potrà sostenere l'esame**;

  - Esecuzione dell'applicazione
    - In questa fase si verifica il funzionamento dell'applicazione e il rispetto delle specifiche;

  - Analisi del codice sorgente
    - Può essere richiesto di spiegare parti del codice e apportarvi semplici modifiche.

- Nel caso in cui la valutazione del progetto sia sufficiente, ma la valutazione complessiva dell'esame non risulti essere sufficiente, il progetto può essere mantenuto. Viene comunque data la possibilità di sottometterne una nuova versione, qualora ci sia spazio per aumentarne la valutazione.

FUNZIONI DI UTILITÀ

- Nel caso si utilizzino i file, una documentazione di alto livello delle funzioni necessarie per leggere e scrivere file a blocchi possono essere visualizzate al seguente indirizzo:
  - https://www.programiz.com/c-programming/c-file-input-output