

Progetto “*Text Express*”

## Descrizione sintetica

Realizzare un sistema client-server che consenta a più utenti contemporaneamente di giocare a una versione semplificata del gioco text express<sup>1</sup>, condividendo lo stesso insieme di lettere. Si utilizzi il linguaggio C su piattaforma UNIX. I processi comunicano tramite socket TCP. Corredare l’implementazione di adeguata documentazione.

## Descrizione dettagliata

Il sistema deve gestire una partita con più giocatori. In ogni partita vengono scelte 6 lettere casuali, di cui 2 vocali e 4 consonanti (escluse q,w,x,y,k,j,h). Compito del giocatore è quello di formare quante più parole possibili (costituite da un minimo di 4 lettere) utilizzando solo le lettere estratte. A differenza del gioco originale, ogni lettera estratta può essere utilizzata anche più di una volta: se vengono estratte le lettere a,c,e,m,p,r la parola “accampare” è accettabile anche se le lettere a e c vengono ripetute più volte. Gli accenti dovranno essere eliminati dalle parole accentate: per cui alla parola “più” corrisponderà la stringa “piu”.

Il giocatore invia le parole formate al server utilizzando il programma client. Ogni volta che il giocatore invia una nuova parola il server prima di assegnare un punteggio verifica se la parola è già stata inviata da un altro giocatore e se è presente sul dizionario<sup>2</sup>. Per ogni parola nuova scoperta viene assegnato un punteggio di 2 punti per le consonanti e 1 punto per le vocali. In qualunque momento, un giocatore può collegarsi ad una partita in corso oppure abbandonare la partita a cui sta partecipando.

Non c’è un limite a priori al numero di giocatori che partecipano contemporaneamente ad una partita. Le partite hanno una durata massima che può essere decisa all’avvio del server come opzione sulla riga di comando.

**Il client** consente all’utente di collegarsi ad un server di gioco, indicando tramite riga di comando il nome o l’indirizzo IP di tale server e la porta da utilizzare. Una volta collegato ad un server, l’utente dovrà fornire un nickname. Quindi l’utente riceverà le lettere della partita in fase di svolgimento, il tempo mancante al termine della partita, l’elenco delle parole che sono state già individuate e il numero di quelle ancora da individuare (contandole dal dizionario). A quel punto, l’utente può ripetutamente proporre parole che saranno inviate al server per controllare se usano solo le lettere previste e se appartengono al dizionario. In caso affermativo, il punteggio dell’utente sarà aggiornato.

Il client deve consentire all’utente durante la partita di sapere qual è il suo punteggio e quale quello degli altri giocatori in gara.

**Opzionale:** fare in modo che, se un giocatore individua una nuova parola, gli altri giocatori vengano immediatamente notificati e vedano tale parola.

**Il server** deve supportare tutte le funzionalità descritte nella sezione relativa al client.

All’avvio del server, è possibile specificare tramite riga di comando la porta TCP sulla quale mettersi in ascolto. Il server è di tipo concorrente, ovvero è in grado di servire più client simultaneamente.

Il server può utilizzare il comando grep per verificare la correttezza delle soluzioni nel file del dizionario a condizione che non si utilizzino le funzioni system, popen e simili ma si utilizzino esclusivamente le system call argomento del programma del corso, come fork, exec, dup e dup2.

Al termine del tempo stabilito, il server invia ai client la classifica con il punteggio di tutti i giocatori.

Durante il suo regolare funzionamento, il server effettua il logging delle attività principali in un file apposito. Ad esempio, memorizza la data e l’ora di connessione dei client e il loro nome simbolico (se disponibile, altrimenti l’indirizzo IP), e la data e l’ora di creazione e di terminazione delle partite.

**Regole generali.** Il server ed il client vanno realizzati in linguaggio C su piattaforma UNIX/Linux. Le comunicazioni tra client e server si svolgono tramite socket TCP. Oltre alle system call UNIX, i programmi possono utilizzare solo la libreria standard del C. Sarà valutato negativamente l’uso di primitive non coperte dal corso (ad es., code di messaggi) al posto di quelle studiate. Il server non deve inviare alcun output su standard

<sup>1</sup> Accessibile gratuitamente all’indirizzo <http://www.it.zylom.com/game/161/text-express.html>

<sup>2</sup> Reperibile all’indirizzo <http://people.na.infn.it/mfaella/Didattica/Labos/2008/dizionario.zip>

output, non deve ricevere nessun input da standard input e può inviare output su standard error solo in caso di terminazione.

## Relazione

Il progetto va accompagnato da una relazione che contenga almeno le seguenti sezioni:

1. Una guida d'uso per il server e per il client, che illustri le modalità di compilazione e d'uso dei due programmi.
2. Una sezione che illustri il protocollo al livello di applicazione utilizzato nelle comunicazioni tra client e server (non il protocollo TCP/IP!).
3. Una sezione che descriva i dettagli implementativi giudicati più interessanti (con particolare riferimento alle system call oggetto del corso), eventualmente corredati dai corrispondenti frammenti di codice.
4. In appendice, la relazione deve riportare il codice sorgente integrale del progetto.

Orientativamente, la relazione dovrebbe constare di almeno 10 pagine, esclusa l'appendice. Indicare sulla copertina della relazione il numero assegnato al gruppo e i componenti del gruppo.

## Consegna del progetto

Entro la data prescelta per lo scritto finale, vanno consegnati al docente il progetto e la relazione. Il progetto e la relazione vanno inviati all'indirizzo [faella.didattica@gmail.com](mailto:faella.didattica@gmail.com) in un archivio compresso in formato zip o rar. La relazione va consegnata al docente *anche* in formato cartaceo. Durante l'esame orale, il client ed il server verranno testati, eseguendoli su due o più macchine diverse.