

L'applicazione distribuita di rete (Client + Server) in oggetto è stata sviluppata in ambiente Linux nativo ed in particolare con la distro Fedora 36 Workstation. Il codice sorgente implementato con l'utilizzo dell'IDE Atom (versione 1.60) durante la fase di testing tramite shell testuale Bash (versione 5.1.16) è stato compilato da terminale mediante il comando 'gcc' (versione 12.2.1 20220819) e le seguenti opzioni '-Wall -Wextra -Werror -pedantic' al fine di garantirne la migliore correttezza e compatibilità possibile con lo standard ISO C del linguaggio C stesso. Entrambi i programmi possono essere eseguiti per comunicare con lo stesso sistema su cui sono in esecuzione creando una connessione localmente all'indirizzo IPv4 del localhost ovvero '127.0.0.1', oppure si può sfruttare il programma Client per connettersi e comunicare anche con un Server remoto, se il suo indirizzo viene correttamente specificato da terminale all'avvio del programma stesso (e.g. `prezzemolo.polito.it` `130.192.9.131`). Per ultima è sempre necessaria anche una porta di riferimento (e.g. `10010`).

Lo scopo dell'applicazione illustrato all'utente è di indovinare, attraverso una serie di tentativi che sono minimo 6 di default e fino ad un massimo di 10 qualora l'utente lo indichi all'avvio del Server, la parola del giorno composta da 5 caratteri alfabetici e scelta da un elenco di parole casuali di senso compiuto definite all'interno del programma Server.

Il server se pronto ed avviato sarà sempre in ascolto ed in attesa di una nuova connessione da parte del Client, l'unico modo per interrompere

veramente la ciclicità in funzione del Server è sfruttare il comando 'ctrl+C' direttamente dal terminale che ne forza prematuramente l'interruzione.

Una volta eseguito il programma Server, questo controlla gli input dati dall'utente da riga di comando e nel caso lo avverte sul corretto modo d'uso dell'applicazione e ne evidenzia gli errori.

Altrettanti controlli degli input da riga di comando saranno verificati anche dal Client al suo avvio.

Successivamente, se sarà stata correttamente stabilita una connessione tra Server e Client, il Server grazie alla funzione 'write' scriverà il messaggio di benvenuto sulla socket che sarà in seguito letto con l'omonima funzione 'read' dal Client ed infine il Server resterà in attesa di ricevere il successivo messaggio in risposta da parte del Client e il Client dopo aver contattato ulteriormente il Server sarà anch'esso in attesa di ricevere un riscontro dato dal precedente messaggio inviato.

Questa condizione di attendere, controllare semanticamente e sintatticamente i messaggi scambiati ed infine rispondere nuovamente si protrae per tutta la durata della connessione tra Server e Client.

Dal lato del client si invita l'utente ad inserire una parola di 5 caratteri oppure a rinunciare ad indovinare con il comando 'QUIT' appositamente proposto e definito dal protocollo di comunicazione dell'applicazione distribuita.

Si entra quindi in un ciclo che darà la possibilità all'utente di continuare a provare ad indovinare la parola del giorno finché avrà ancora almeno un tentativo a disposizione, ma solamente se avrà

precedentemente digitato una parola non considerata illegale per il Server, dato che ad ogni inserimento dell'utente il Server controlla sempre che la parola digitata contenga unicamente caratteri di codifica ASCII, appartenga alla classe Alpha definita nella libreria standard '<ctype.h>' del linguaggio C e sia anche della lunghezza adeguata ovvero precisamente 5 caratteri.

Il Server dunque, dopo aver terminato i controlli successivi ad ogni inserimento dell'utente, se non sono andati a buon fine ha il compito di avvisare il Client con il comando opportuno e l'ausilio della parola chiave del protocollo 'ERR', che identifica gli errori, per far immediatamente interrompere il ciclo e chiudere la connessione in corso non appena uno dei requisiti sopracitati non venisse più rispettato ed allo stesso modo il Server comunica anche con altre parole chiave ovvero: 'OK PERFECT', 'OK' ed 'END' definite dal protocollo di comunicazione per informare il Client rispettivamente che la parola del giorno è stata indovinata ('OK PERFECT'), che ci sono ancora dei tentativi rimanenti ('OK') oppure in ultima analisi che sono terminati i tentativi disponibili ('END').

Il Client scambia i suoi messaggi col Server mediante due parole chiave che sono: 'WORD' per dimostrare la volontà dell'utente di fare un tentativo di identificazione della parola del giorno e 'QUIT' per indicare che l'utente vuole terminare la comunicazione.