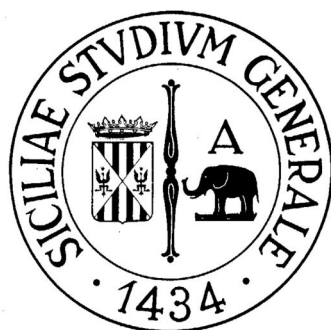


---

## - PROGETTO CLIENT IRC -

Irene Baldacchino (O46001182)  
Pierluigi Gaglio (O46001301)

Università degli studi di Catania  
anno accademico 2017/18



---

## INTRODUZIONE

Il progetto propostoci, chiede di realizzare un Client IRC utilizzando la socket TCP (SOCK\_STREAM). Il progetto terminato include:

- 1) Main Thread, il quale gestisce il file di configurazione, la connessione alla socket, l'inserimento delle credenziali per il login al server e un menù;
  - 2) Thread di lettura, il quale permette la lettura dalla socket e di mantenere attiva la connessione mediante il comando PING;
  - 3) Executer, il quale permette di eseguire comandi da shell tramite ricezione di un particolare Keyword (PIEXE\*).
-

---

## SVOLGIMENTO

La prima operazione implementata all'interno del client è quella relativa all'apertura di un file (`file_configurazione.txt`) che presenta due elementi: la porta e l'indirizzo del server. Questi vengono estratti dal file e memorizzati all'interno delle variabili "indirizzo" e "porta" al fine di connettersi al sever. Stabilita la connessione, è stata implementata la funzione "Inserisci\_Credenziali" al fine di inserire i dati: USER, NICK e PASS necessari per il login al server.


Essi sono stati implementati utilizzando la funzione `sprintf()`, in modo tale da assemblare automaticamente il comando da inviare al server.

L'operazione necessaria per proseguire è quella relativa alla creazione del thread `thread_lettura` con lo scopo di leggere "in background" dalla socket.

Il tutto si conclude con la presentazione di un menù che permette all'utente di scegliere tra le varie operazione disponibili:

- 1) Joina in un canale: permette di entrare all'interno di una stanza inserendo soltanto il nome di quest'ultima (omettendo #).
  - 2) Lascia il canale: permette di uscire dalla stanza corrente.
  - 3) Messaggio privato: permette di inviare un messaggio ad un utente presente nel canale corrente.
  - 4) Messaggio pubblico: permette di inviare un messaggio nel canale corrente che risulterà visibile a tutti.
  - 5) Chiudi IRC: chiude la connessione con il server.
-

---



Tutti i comandi da inviare alla socket sono stati composti mediante la `sprintf()` ed inviati mediante la `write()`. Il `thread_lettura` permette di eseguire diverse operazioni in base al comando ricevuto leggendo senza interruzione dalla socket. Possiamo distinguere quindi diversi casi:


1) In caso di lettura del comando PING: il client invia automaticamente il comando PONG così da non perdere la connessione con il server;

2) In caso di lettura del comando PRIVMSG: estrae dalla stringa, la quale contiene il messaggio ricevuto, il nome dell'utente da cui si riceve il messaggio che è compreso tra ':' e '!'. Per estrarre il nome dell'utente è utilizzata la funzione `strchr()` che ritorna il puntatore alla prima occorrenza del carattere indicato. Successivamente viene copiata in una variabile `utente_1` la stringa interessata dopo ':'; viene sostituito così il primo carattere con uno spazio al fine di spezzare la stringa in base ai caratteri '!' ' ' (spazio). Infine stampa da chi è stato inviato il messaggio ed il suo contenuto.

Nell'eventualità in cui questo legga oltre il comando PRIVMSG anche la Keyword "PIEXE\*", automaticamente viene estratto il canale in cui è stato inviato (utilizzando il medesimo algoritmo di estrazione precedente) ed avvia la funzione `executer()`.

La funzione `executer()` permette di eseguire un comando direttamente su shell mediante la funzione `system()`, la quale ha come parametro un comando del tipo "comando > file.txt", che viene eseguito direttamente su shell salvando il suo contenuto (l'Output) su un file di testo.

---



---

Di conseguenza verrà letto ciò che è contenuto nel file ed inviato come messaggio privato all'utente mittente. Diversamente, se viene inviato come comando da eseguire "cd /directory", verrà eseguita la funzione `funzione_comando_cd`. Questa mediante `getcwd()` restituirà la directory corrente e tramite `strcat()` concatenerà quest'ultima con /directory. Infine con il comando `chdir()` avverrà il cambiamento.

---