<https://github.com/lucacorbucci/ChatterBox/blob/master/Testo%20Progetto/chatterbox17.pdf>

https://github.com/Errore418/ProgettoSO1718/blob/master/doc/Relazione.pdf

**RELAZIONE SISTEMI OPERATIVI 2020/2021**

Docente: Prof. Francesco Quaglia



Macroarea Ingegneria

Anuar Elio Magliari, 0267412

Michele Tosi, 0253591

**Sommario**

**Introduzione**

Il modulo di Sistemi Operativi prevede lo svolgimento di un progetto da sviluppare utilizzando il linguaggio C in ambiente Unix. Questo documento descrive la struttura complessivia del progetto. Esso ha come scopo la realizzazione di un servizio “talk” via internet gestito tramite server.

**Descrizione generale**

Il servizio tiene traccia di un elenco di client pronti ad iniziare una "conversazione". I client (residenti, in generale, su macchine diverse), dopo essersi connessi al servizio acquisiscono l'elenco e lo mostrano all'utente il quale potrà collegarsi a un altro utente disponibile iniziando cosi' la conversazione. Due client collegati tra loro permetteranno ai relativi utenti di scambiarsi messaggi finche' uno dei due non interrompe la conversazione, e saranno indisponibili ad altre conversazioni. Al termine di una conversazione i client ritorneranno disponibili fino a che non si scollegano dal server.

La comunicazione tra client e server avviene tramite socket di tipo AF\_INET. Il server al suo interno implementa un sistema multi-thread in grado di gestire concorrentemente sia nuove connessioni da nuovi client che gestire le richieste su connessioni già stabilite.

Un client può collegarsi al server in ascolto identificandosi e rendendosi disponibile ad altri client. Su richiesta può intraprendere una conversazione

con i client disponibili nel display della tabella, se e solo se viene accettata da quest’ultimo. La comunicazione viene interrotta a seguito del messaggio “quit” oppure con il segnale “SIGINT” rendendosi di nuovo disponibili. Possono scollegarsi dal server solo nel momento in cui non hanno nessuna comunicazione in corso.

Il server può gestire fino ad un massimo di connessioni concorrenti stabilito dalla “define” PENDING, se tale numero viene superato, il client rimarrà in attesa di un posto disponibile o altrimenti si scollegherà. Il server può essere terminato solo se viene chiuso il suo terminale oppure tramite un segnale di “SIGKILL”.

**Gestione Dati**

* Discussione sulle strutture dati: strutture e liste
* Gestione della memoria dinamica con malloc
* Discussione sulla gestione della concorrenza dei dati (semafori)

**Breve descrizione sulla logica dei programmi**

//cosa fanno le varie funzioni

**Server**

* + client\_online: fornisce una tabella dei client disponibili ad intraprendere una conversazione, in caso di nessuna disponibilità il thread assegnato rimane in attesa
  + comunicazione: permette una conversazione tra i client tramite socket

**Gestione segnali**

**SIGINT**

**Esempi di codice per testing**