

Applicazione che implementa una rete peer-to-peer per la condivisione di dati relativi alla pandemia COVID-19. L'applicazione è formata da un discovery server (DS) che si occupa di mettere in comunicazione i diversi peer quando essi si vogliono connettere o disconnettere alla rete. Ogni peer poi ha la possibilità di inserire dati, richiederli agli altri peer ed effettuare delle elaborazioni su di essi.

## CONTENUTO PROGETTO

### - ds.c

Contiene il codice per eseguire il compito del discovery server.

Attraverso un socket UDP accetta messaggi di connessione e disconnessione da parte dei peer mantenendo aggiornata la lista e richieste di blocco e sblocco della rete per mantenerla statica durante le operazioni di flood.

Accetta inoltre semplici comandi da tastiera per visualizzare informazioni sulla rete o per disconnettersi.

Inoltre mantiene aggiornati i peer sulla data da considerarsi corrente per l'inserimento dei dati.

### - peer.c

Contiene il codice per permettere ai peer di connettersi e disconnettersi dalla rete, aggiungere dati, condividerli ed elaborarli.

Attraverso un socket UDP comunica con il server e con gli altri peer per scambiare messaggi di coordinazione.

Attraverso un socket listener TCP accetta richieste di dati sensibili, scambiandoli sempre attraverso connessioni TCP.

Sono disponibili dei comandi da terminale per inserire dati e per richiederne elaborazioni.

### - util/

Contiene i seguenti file di utilità con i relativi header:

date.c	file per la gestione e manipolazione di strutture 'Date'.
msg.c	file per la gestione di socket udp e tcp e relativi invii e ricezioni di messaggi.
peer_file.c	file per mantenere e modificare la struttura PeerList, lista di PeerElement, che rappresenta lo stato della rete.
neighbors.c	file che definisce la struttura Neighbors per la gestione dei vicini.
util_c.c	file che contiene funzioni generiche per il peer.
util_s.c	file che contiene funzioni generiche per il discovery server.
util.c	file che contiene funzioni generiche.

Nei file.h sono presenti le definizioni delle costanti e commenti che descrivono le diverse funzioni.

### - makefile

### - exec.sh

Compila il progetto attraverso il makefile e poi avvia il ds e 5 peer in locale.

### - commands.txt

Contiene dei comandi di default per testare i peer

### - data/

Contiene i registri dei peer secondo uno schema ad albero: data/port/type/aggr/date.txt

- port rappresenta la porta del peer a cui appartengono i registri (es. 5001)
- type rappresenta il tipo di dati salvati (es. 't' / 'n')
- aggr rappresenta se si tratta di un registro (entries) o di aggregazioni (elabs)
- date nel formato yyyy\_mm\_dd

### - support/

Contiene i seguenti file di documentazione:

codici.txt	sono presenti i dettagli degli header usati per la comunicazione
ds.txt	descrizione in metacodice del comportamento del DS
peer.txt	descrizione in metacodice del comportamento dei peer
readme.pdf	questo file