

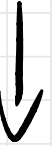
FUNZIONI/TIPS

UDP / Server

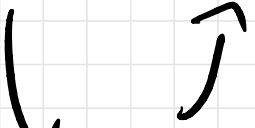
socket()



Bind()



recvfrom()



sendto()



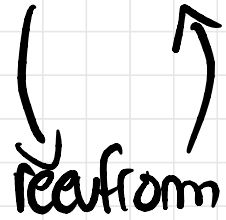
close()

UDP / Client

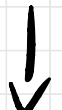
socket()



sendto()



recvfrom



close()

Cosa fanno queste funzioni ??

Socket() → Creare la socket, Restituire descrittore di file.

- `socket(AF_INET, SOCK_DGRAM, 0)`

`bind()` → Assegna l'indirizzo locale a una socket.

- `bind(sockfd, (struct sockaddr*)&local_addr, sizeof(local_addr));`

↙ descrittore
a cui fare la
bind

↙ indirizzo a cui fare
la bind()

↙ dimensione in byte
di LOCAL_ADDR;

`recvfrom()` → Ricevere dati da una socket non orientata alla
connessione. (È necessario specificare il descrittore di file della socket ecc....)

- `recvfrom(sockfd, buffer, sizeof(buffer), 0, (struct sockaddr*)&addr, &size)`

↙ `addr, &size`
`sizeof(addr)`

`sendto()` → Inviare i dati ad una socket

`char* response[Buf.size];`

- `sendto(sockfd, response, strlen(response), 0, (struct sockaddr*)&addr, size)`

↙ dimensione buffer
di risposta.

`close()` → chiudere la socket()

- `close(sockfd);`

TEP/Server

TEP/Client

Compto 06/07/2021

Gli studenti del corso di Reti di Calcolatori del DMI – UNICT, hanno deciso, di loro spontanea volontà, di realizzare il gioco dell'impiccato tra macchine virtuali.

Due macchine client si sfideranno tra di loro al fine di indovinare la parola segreta presente in una macchina server. **Ogni macchina client avrà a disposizione 3 vite.** Il conteggio delle vite sarà tenuto dal server. Ad ogni turno:

- Il server prima invierà al client lo stato del gioco: lettere indovinate, lettere mancanti e la struttura della parola¹.
- Successivamente, ogni client ad ogni turno avrà la possibilità solo di scrivere da riga di comando **o una lettera o dare la risposta completa.** In caso di errore, perderà una vita. Il server notificherà allo specifico client l'esito del turno.
- Se il giocatore di turno perde tutte le vite, dovrà terminare l'esecuzione del programma².

Il server dovrà memorizzare in una **struct client** il nome del giocatore, l'indirizzo IP e la porta del client. Notare che, massimo possono giocare solo 2 client a partita e il gioco inizierà solo quando il server avrà registrato (struct client) i due giocatori³.

Il server sceglierà randomicamente una stringa da utilizzare per il gioco dell'impiccato tra quelle presenti in un array bidimensionale.

Il server sceglierà il client che inizierà la partita con modalità a scelta libera dello studente (random, il primo che si è registrato, ecc.)

CI

Socket()

(memorizza
family
port)

• Prende da input

sendto()

il nome per identificazione

recvfrom()

Riceve la parola da indovinare e vite

sendto() manda parola / lettera

Server