Programimi

I Soket ne gjuhen C

You can reach me on [www.facebook.com/flori12](http://www.facebook.com/flori12)

 Flor\_iano@hotmail.com

Greetz too INFOSEC Institute and Str0ke

Ky eshte nje tutorial ne programimin e soket ne gjuhen C per sistemet e Linux-it.”Linux” sepse kodet qe jane dhene me poshte do te punojne vetem ne sistemet Linux dhe Jo ne Sistemet Windows.Programimi I soket api ne Windows do te quhet winsock dhe per kete do te kemi nje tjeter tutorial .

**Cfare eshte nje Soket?**

**--Nje pershkrues qe lejon aplikacionet te lexojne dhe te shkruajn nga/ne network.**

**--I ngjashem per network I/O sic jane skedaret I/O.**

**--Klientet dhe serverat komunikojn duke lexuar/shkruar nga/ne pershkruesit e socketave.**

**Structura e soketave te adresave te Internetit.**

**struct sockaddr\_in {**

**unsigned short sin\_family; /\* Familja e adreses (gjithmone AF\_INET) \*/**

**unsigned short sin\_port; /\* Numri I portit ne network \*/**

**struct in\_addr sin\_addr; /\* Adresa IP ne network \*/**

**unsigned char sin\_zero[8]; /\* mbushja ne sizeof(struct sockaddr) \*/**

**};**

**struct sockaddr {**

**unsigned short sa\_family; /\* Familja e protokollit \*/**

**char sa\_data[14]; /\* Adresat e datave \*/**

**};**

**/\*Struktura e hyrjeve te DNS \*/**

**struct hostent {**

**char \*h\_name; /\* Emri zyrtar I domainit \*/**

**char \*\*h\_aliases; /\* Domain alias\*/**

**int h\_addrtype; /\* Tipi I adreses HOST (AF\_INET) \*/**

**int h\_length; /\* Gjatesia e nje adrese, ne byte \*/**

**char \*\*h\_addr\_list; /\* Nderprerja e vektoreve ne structure in\_addr \*/**

**};**

**#define h\_addr h\_addr\_list[0] /\* Hyrja e pare ne h\_addr\_list \*/**

– **gethostbyname() :** Kerkimi duhet te jete nje domain.

– **gethostbyaddr() :**  Kerkimi duhe te jete nje Address IP.

Sockets jane pjeset virtual te cdo lloj komunikimi ne network I bere nga 2 hoste mbi nje lidhje.

Per shembull kur ti shkruan [www.google.com](http://www.google.com) ne shfletuesin tend, ai hap nje socket dhe me pas ben lidhje me google.com per te krahasuar me faqen dhe me pas ju a tregon juve.E njejta gje ndodh dhe ne cdo cate si gtalk ose skype.Cdo komunikim I rrjetit shkon percart nje socketi.

Socket API ne Linux eshte e ngjashme me soket e bsd/unix nga I cili ka evoluar.Edhe pse

Ne cdo kohe api eshte bere I ndryshem ne disa sisteme.Dhe tani standarti me I ri eshte posix sockets api e cila eshte e njejte me bsd sockets.

Per te kuptuar kete tutorial ti duhet te kesh pak njohuri ne gjuhen C dhe ne pointerat.Ju duhet te keni nje kompilues gcc I cili eshte instaluar ne sistemin Linux.Nje IDE se bashke me gcc do te ishte shume mir.Une do te ju rekomandoja geany sepse ju mund te editoni dhe te drejtoni nje program ne te pa bere shume konfigurime.Ne ubuntu ti mund te shkruash ne terminal apt-get install geany.Ne te gjithe tutorialin jane kode te cilat demostrojne disa koncepte.Ti mund te drejtosh keto kode ne geany dhe te shifni menjehere rezultate per ti kuptuar me mire konceptet..

**Krijimi I nje Socketi**

Gjeja e pare qe duhet te bere eshte te krijosh nje socket.Funksioni socket eshte sipas kodit me poshte:

|  |  |
| --- | --- |
| 123456789101112131415 | #include<stdio.h>#include<sys/socket.h>  int main(int argc , char \*argv[]){    int socket\_desc;    socket\_desc = socket(AF\_INET , SOCK\_STREAM , 0);         if (socket\_desc == -1)    {        printf("Nuk mund te krijoj nje socket");     }         return 0;} |
|  |  |

Funksioni socket() krijon nje socket dhe kthen nje pershkrues e cila mund te perdoret ne funksionet e tjera. Kodi me siper do te krijoj nje socket me keto veti.

Famila e Adresave- AF\_INET (Vetem per versionin Ipv4)

Tipi - SOCK\_STREAM (Kjo do te thote qe eshte socket I tipit te TCP)

Protokolli - 0 [ ose IPPROTO\_IP Protokolli IP]

Me pas mund te duhet te lidhemi me nje server duke perdorur socket.

Ne mund te lidhemi me faqen www.google.com

**Shenim**

Pervec tipit SOCK\_STREAM eshte nje tjeter tip I cili quhet SOCK\_DGRAM I cili lidhet me protokollin UDP.Ky tip I soket quhet nje jo-lidhje e socketave.Ne kete tutorial do ti permbahemi tipi SOCK\_STREAM ose socketave te TCP.

**Lidh socketin ne server**

Ne mund te lidhemi ne nje remote server ne nje port te caktuar.Keshtu na duhen 2 gjera, adresa IP dhe numri I portit me te cilen do lidhemi.Per tu lidhur ne nje remote server neve na duhet te kryejme disa veprime.E para eshte te krijojme nje strukture me vlera te qarta.

struct sockaddr\_in server;

Shikoni me poshte kodin.

|  |  |
| --- | --- |
| 12345678910111213141516 | // IPv4 AF\_INET sockets:struct sockaddr\_in {    short            sin\_family;   // AF\_INET, AF\_INET6    unsigned short   sin\_port;     // htons(3490)    struct in\_addr   sin\_addr;     // shikon strukturen in\_addr, poshte    char             sin\_zero[8];  // Bejeni 0 nese doni.}; struct in\_addr {    unsigned long s\_addr;          // Ngarko inet\_pton()}; struct sockaddr {    unsigned short    sa\_family;    // Familja e adreses, AF\_xxx    char              sa\_data[14];  // 14 byte adresa e protokollit}; |

Struktura sockaddr\_in ka nje pjestar te quajtur sin\_addr te tipit in\_addr e cila ka nje s\_addr e clia nuk eshte asgje vecse nje fjali e rendesishme.Ajo permban Adresat IP ne nje format te gjate

Funksioni inet\_addr eshte nje funksion shume I dobishem I cili konverton nje IP address ne nje format te gjate. Kjo eshte menyra si e bejme ne:

|  |  |
| --- | --- |
| 1 | server.sin\_addr.s\_addr = inet\_addr("74.125.235.20"); |

Keshtu ju duhet te dini adresen e IP-ise te serverit me te cilin do lidheni.Ketu ne perdorem adresen IP te google.com si nje shembull.Pak me von ne duhet te shohim si te gjejm adresen IP te nje domaini te dhene nga perdoruesi ose I caktuar ne program :D.

Gjeja e fundit qe na duhet eshte funksioni connect().Ky do nje socket dhe strukturen sockaddr per tu lidhur.Me poshte jepet kodi per shembullin ne fjale.

|  |  |
| --- | --- |
| 123456789101112131415161718192021222324252627282930 | #include<stdio.h>#include<sys/socket.h>#include<arpa/inet.h> //inet\_addr int main(int argc , char \*argv[]){    int socket\_desc;    struct sockaddr\_in server;         //Krijimi I soket    socket\_desc = socket(AF\_INET , SOCK\_STREAM , 0);    if (socket\_desc == -1)    {        printf("Socketi nuk mund te krijohet");    }             server.sin\_addr.s\_addr = inet\_addr("74.125.235.20");    server.sin\_family = AF\_INET;    server.sin\_port = htons( 80 );     //Lidhu ne nje remote server    if (connect(socket\_desc , (struct sockaddr \*)&server , sizeof(server)) < 0)    {        puts("Gabim ne lidhje");        return 1;    }         puts("Lidhja u krye");    return 0;} |

Kjo nuk mund te behet me e thjeshte se kaq.Krijohet nje soket dhe me pas lidhet me remote server.Nqs do te nisni programin ai duhet te jape “I lidhur”.Ti duhet te nisesh programin me privilegje te root.Nqs jep gabim ne lidhje atehere ti nuk ke lidhje me Internetin ose nuk e ke nisur programin me privilegje te root.

Per te nisur programin ne root ti duhet te shkruash : sudo ./a.out

Provoni te lidheni ne nje port te ndryshem nga 80 dhe keshtu juve nuk do te ju mundesohen lidhja e cila jep se porti nuk eshte I hapur per lidhje.
Ok, tani ne jemi te lidhur.Le te bejme dicka tjeter, te transmetojme disa data ne remote server.

**Lidhjet jane te pranueshme vetem ne soket e TCP**

Koncepti I “lidhjes” aplikon tipin e soketit -> SOCK\_STREAM/TCP.Lidhja nepermjet ketij streami eshte I besueshem.

Mendoni sikur ky te ishte nje pip I jo nderprere nga data te tjera.

Soketat e ndryshem sic jane UDP, ICMP, ARP nuk kane kete koncept te lidhje.Ato ja jo-lidhes I bazuar ne komunikim.E cila do te thote ti mund ti cosh data dhe te marresh paketa nga cdokush.

**Transfero data mbi rrjet**

Funksioni send() do te thjeshtesoje pune per te transferuar data,Ky do te pershkruesi te soket, te dhenat qe do transferoje dhe madhesine e tij.
Me poshte ju jepet nje shembull shume I qarte per transferimi e disa te dhenave tek ip a google.com:

|  |  |
| --- | --- |
| 12345678910111213141516171819202122232425262728293031323334353637 | #include<stdio.h>#include<string.h>    //strlen#include<sys/socket.h>#include<arpa/inet.h> //inet\_addr int main(int argc , char \*argv[]){    int socket\_desc;    struct sockaddr\_in server;    char \*message;         //Krijo soketin    socket\_desc = socket(AF\_INET , SOCK\_STREAM , 0);    if (socket\_desc == -1)    {        printf("Lidhja nuk mund te kryhet”);    }             server.sin\_addr.s\_addr = inet\_addr("74.125.235.20");    server.sin\_family = AF\_INET;    server.sin\_port = htons( 80 );     //Lidhu ne remote server    if (connect(socket\_desc , (struct sockaddr \*)&server , sizeof(server)) < 0)    {        puts("Gabim ne lidhje");        return 1;    }         puts("I lidhur\n");         //Duke transferuar disa data     message = "GET / HTTP/1.1\r\n\r\n";    if( send(socket\_desc , message , strlen(message) , 0) < 0)    {        puts("Dergimi I te dhenave deshtoi");        return 1;    }    puts("Te dhenat u derguan\n");         return 0;} |

Ne shembullin me larte, ne fillim te lidhemi ne nje adrese ip dhe me pas I cojme mesazhin te tipi string "GET / HTTP/1.1\r\n\r\n" serverit.

Ky mesazj aktualish eshte nje komand e protokollit http per te provuar faqen kryesore te nje webi.Tani qe ne kemi cuar disa te dhena, eshte koha te marrim reply nga server.Keshtu le a bejme kete program te devotshem ne kete menyre.

**Shenim**

Kur transferojm te dhena tek nje soketi ne jemi duke shkruar te dhena ne kete soket.Kjo eshte e ngjashme si te shkruash te dhena ne nje skedar.Keshtu ti mund te perdoresh funksionin write() per te derguar te dhenat tek nje soket I deshiruar.

Me vone ne kete tutorial ne do te duhet te perdorim funksionin write() per te derguar te dhena.

|  |  |
| --- | --- |
| 12345678910111213141516171819202122232425262728293031323334353637383940414243 | #include<stdio.h>#include<string.h>    //strlen#include<sys/socket.h>#include<arpa/inet.h> //inet\_addr int main(int argc , char \*argv[]){    int socket\_desc;    struct sockaddr\_in server;    char \*message , server\_reply[2000];         //Krijimi I soket    socket\_desc = socket(AF\_INET , SOCK\_STREAM , 0);    if (socket\_desc == -1)    {        printf("Nuk mund te krijohet soketi");    }             server.sin\_addr.s\_addr = inet\_addr("74.125.235.20");    server.sin\_family = AF\_INET;    server.sin\_port = htons( 80 );     //Lidhja e soketit me remote server    if (connect(socket\_desc , (struct sockaddr \*)&server , sizeof(server)) < 0)    {        puts("Lidhja nuk u krye ");        return 1;    }         puts("Lidhja u be.\n");         //Dergo disa te dhena    message = "GET / HTTP/1.1\r\n\r\n";    if( send(socket\_desc , message , strlen(message) , 0) < 0)    {        puts("Dergimi deshtoi");        return 1;    }    puts("Te dhenat u derguan me sukses\n");         //Marrja e nje pergjigjeje nga serveri    if( recv(socket\_desc, server\_reply , 2000 , 0) < 0)    {        puts("Marrja deshtoi");    }    puts("Marrja u krye me sukses\n");    puts(server\_reply);         return 0;} |

**Marrje e te dhenave ne soket**

Funksioni recv() eshte perdorur per te marre te dhenat nga soketi.Ne shembullin me poshte ne do te duhet te dergojme te njejtin mesazh si tek shembulli mer lart dhe te marrim nje pergjigje nga serveri

Ky eshte prodhimi I kodit me lart :

Connected

Data Send

Reply received

HTTP/1.1 302 Found

Location: http://www.google.co.in/

Cache-Control: private

Content-Type: text/html; charset=UTF-8

Set-Cookie: PREF=ID=0edd21a16f0db219:FF=0:TM=1324644706:LM=1324644706:S=z6hDC9cZfGEowv\_o; expires=Sun, 22-Dec-2013 12:51:46 GMT; path=/; domain=.google.com

Date: Fri, 23 Dec 2011 12:51:46 GMT

Server: gws

Content-Length: 221

X-XSS-Protection: 1; mode=block

X-Frame-Options: SAMEORIGIN

<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">

<TITLE>302 Moved</TITLE></HEAD><BODY>

<H1>302 Moved</H1>

The document has moved

<A HREF="http://www.google.co.in/">here</A>.

</BODY></HTML>

Ne mund te shohim se kjo pergjigje eshte derguar nga serveri.Ky format shfaqet ne formen HTML, sepse ne te vertete ajo eshte HTML.Google.com dergoi permbajtjen e faqes qe ne kerkuam.Shume e thjeshte!

**Shenim**

Kur merr te dhenat ne nje soket, ne saktesisht po lexojme te dhenat ne po kete soket.Kjo veti eshte e ngjashme si te lexosh te dhenat nga nje skedar.Keshtu ne mund te perdorim funksionin read() per te lexuar te dhenat ne nje soket.Per shembull::

|  |  |
| --- | --- |
| 1 | read(socket\_desc, server\_reply , 2000); |

Tani qe ne kemi marre pergjigjen tone, eshte koha te mbyllim soketin.

**Mbyllja e soketit**

Funksioni close() eshte perdorur per te mbyllur nje soket.Ne duhet te perfshijme librarine <unistd.h> per kete funksion.

|  |  |
| --- | --- |
| 1 | close(socket\_desc); |

Kjo eshte.

**Permbledhje**

Ne shembullin me larte ne mesuam se si te :

1. Krijojm nje Soket

2. Te lidhemi ne nje remote server

3. Te dergojm disa te dhena

4. Marrim Pergjigje

Shfletuesi juaj vepron saktesisht ne kete menyre qe ne momentin kur ti shkon ne nje webpage.Ky tip I soket prezanton aktivitetin e quajtur klienti soket.Nje klient eshte nje aplikacion I cili lidhe me remote system per te marre te dhenat e deshiruara.

Tipi tjeter I soket eshte quajtur server soket.Nje server eshte nje sistem I cili perdor soketat per te marre lidhje te autorizuara nga aplikacioni.Eshte e kunderta e Klientit.Keshtu [www.google.com](http://www.google.com) eshte serveri dhe shfletuesi eshte klienti.Ose ne menyre me te teknike [www.google.com](http://www.google.com) eshte nje server HTTP dhe shfletuesi eshte klienti HTTP

Tani eshte koha te bejme disa detyra te serverit duket perdorur soketat.Por perpara ne do te shkojme me tej ku jane prekur disa subjekte ne raste kur do te ju duhen.

**Shfaq adresen ip te nje hosti te dhene**

Kur lidheni ne nje remote host, eshte e nevojshme te keni adresen IP te tij.Funksioni gethostbyname() eshte perdorur per kete qellim.Ky funksion merr emrin e domainit si nje parameter dhe kthen strukturen e tipit hostent.Kjo strukture ka informacionet IP.Kjo strukture perfshihet te librarine <netdb.h>.Le ti hedhim nje sy kesaj strukture

. h\_addr\_list ka adresat ip.Keshtu ne tani kemi disa kode ku ti perdorim keto funksione.

|  |  |
| --- | --- |
| 12345678910111213141516171819202122232425262728 | #include<stdio.h> //printf#include<string.h> //strcpy#include<sys/socket.h>#include<netdb.h> //hostent#include<arpa/inet.h> int main(int argc , char \*argv[]){    char \*hostname = "www.google.com";    char ip[100];    struct hostent \*he;    struct in\_addr \*\*addr\_list;    int i;             if ( (he = gethostbyname( hostname ) ) == NULL)     {        //gethostbyname deshtuesi        herror("gethostbyname");        return 1;    }//Hedh h\_addr\_list ne in\_addr , qe kur h\_addr\_list ka gjithashtu Adresat ip ne format te gjate.     addr\_list = (struct in\_addr \*\*) he->h\_addr\_list;         for(i = 0; addr\_list[i] != NULL; i++)     {        //Kthe te parin        strcpy(ip , inet\_ntoa(\*addr\_list[i]) );    }         printf("%s u shpetua ne IP : %s" , hostname , ip);    return 0;} |

Output of the code would look like :

www.google.com resolved to : 74.125.235.20

Keshtu kodi me lart eshte perdorur per te gjetur adresa ip te nje domaini te dhene.Me pas adresat ip mund te perdoren per te kryer lidhjen tek nje soket I caktuar.

Funksioni inet\_ntoa() do te konvertoje adresen IP te formatit te gjate ne format me pika.Kjo eshte e kunderta e inet\_addr().

Se gjati ne kemi pare struktura te rendesishme te cilat jane perdorur.Le ti shofim keto:

1. sockaddr\_in – Informacion I lidhjes. I perdorem nga funksioni connect(),send(),recv().

2. in\_addr – Adresa IP ne format te gjate.
3. sockaddr
4. hostent – Adresat ip te nje hosti. I perdorur nga gethostbyname.

Ne pjesen me vone do te shofim se si te krijojm duke perdorur soketat.Serverat jane te kundertat e klientit, ne vend te kesaj duke u lidhur me te tjeret, ato te presin per lidhjet hyrese.

**Serverat Soket**

Ok tani jemi ne gjerat e serverat.Soketat servers operojn ne nje menyre te ndryshme.

1. Krijimi I nje Serveri

 2. Lidhja e nje adresi(dhe porti).

3. Degjimi per lidhjet hyrese.

4. Pranimi I lidhjeve

5. Lexo/Dergo

Ne kemi mesuar se si te hapim dhe te mesojme nje soket gjithashtu dhe ta mbyllim ate.Keshtu ne duhet te bejme lidhjen me funskionin bind().

**Lidh soketin ne nje port**

Funksioni bind() mund te perdoret per tu lidhur me nje soket ne nje “adres dhe port” kombinimi.Ketu duhet struktura sockaddr\_in e ngjasjme me funksionin connect().

|  |  |
| --- | --- |
| 123456789101112131415161718 | int socket\_desc;struct sockaddr\_in server;     //Krijimi I soketsocket\_desc = socket(AF\_INET , SOCK\_STREAM , 0);if (socket\_desc == -1){    printf("Soketi nuk mund te krijohet");}     //Pergatit strukturen sockaddr\_inserver.sin\_family = AF\_INET;server.sin\_addr.s\_addr = INADDR\_ANY;server.sin\_port = htons( 8888 );     //Lidhjaif( bind(socket\_desc,(struct sockaddr \*)&server , sizeof(server)) < 0){    puts("Lidhja deshtoi");}puts("Lidhja u krye"); |

Tani qe lidhja eshte kryer, eshte koha ta bejme soketin te degjoje lidhjet hyrese(connections).Ne lidhim soketin ne nje adres IP dhe ne nje port te caktuar.Duke bere kete ne sigurohemi se te gjitha te dhenat qe jane duke ardhur te drejtohen ne kete port duke u pranuar nga aplikacioni perkates per te kryer kete veprim.Kjo e ben te qarte se 2 soketa nuk mund te lidhen ne te njejtin port.

**Degjo per lidhjet hyrese ne soket**

Mbas lidhjes tek nje soket ne nje port, gjeja tjeter qe duhet te bejme eshte te degjojme per koneksionet e mundshmme.Per kete ne duhet te perdorim soketin ne menyren e degjimit.Funksioni listen() eshte perdorur per te futur soketin ne menyren degjuese.Vetem shtoni rrjeshtat me poshte mbas lidhjes:

|  |  |
| --- | --- |
| 12 | //Degjolisten(socket\_desc , 3); |

Kjo eshte e gjitha.Tani shkojme tek pjesa kryesore pranimin e lidhjeve te reja.

**Prano Lidhjet**

|  |  |
| --- | --- |
| 1234567891011121314151617181920212223242526272829303132333435363738 | #include<stdio.h>#include<sys/socket.h>#include<arpa/inet.h> //inet\_addr int main(int argc , char \*argv[]){    int socket\_desc , new\_socket , c;    struct sockaddr\_in server , client;         //Lidhja e soketit    socket\_desc = socket(AF\_INET , SOCK\_STREAM , 0);    if (socket\_desc == -1)    {        printf("Soketi nuk mund te krijohet");    }         //Pergatit strukturen sockaddr\_in    server.sin\_family = AF\_INET;    server.sin\_addr.s\_addr = INADDR\_ANY;    server.sin\_port = htons( 8888 );         //Lidhja ose Bind,ngurtesimi    if( bind(socket\_desc,(struct sockaddr \*)&server , sizeof(server)) < 0)    {        puts("Lidhja deshtoi");    }    puts("Lidhja u krye");         //Degjo    listen(socket\_desc , 3);         //Pranimi I lidhjeve hyrese    puts("Duke pritur per lidhjet...");    c = sizeof(struct sockaddr\_in);    new\_socket = accept(socket\_desc, (struct sockaddr \*)&client, (socklen\_t\*)&c);    if (new\_socket<0)    {        perror("lidhja deshtoi");    }         puts("Lidhja u aplikua");     return 0;} |

Funksioni accept() eshte perdorur per te pranuar lidhjet hyrese.

**Prodhimii I programit**

Nisja e programit. Ky duhet te shfaqe si me poshte

Lidhja u aplikua

Duke pritur per lidhjet hyrese . . .

Keshtu tani programi eshte duke pritur per lidhjet hyrese ne portin 8888.Mos e mbyll kete program, mbaje te ngarkuar.

Tani klienti mund te lidhet ne kete soket ne portin e caktuar.Do te duhet te perdorim klientin telnet per ta testuar kete.
Hap terminalin dhe shtyp

$ telnet localhost 8888

Ne terminal ti duhet te marresh

Trying 127.0.0.1...

Connected to localhost.

Escape character is '^]'.

Connection closed by foreign host.

Dhe serveri do te shfaqe

Lidhja u aplikua

Duke pritur per lidhjet hyrese...

Lidhja u krye.

Keshtu ne tani mund te shohim se klienti eshte lidhur me serverin.Provo procesin me lart derisa you te beheni me perfekt.

**Merr adresen IP dhe lidhu me klientin**

Ti mund te marresh adresen IP te nje klienti ne nje lidhje nga porti duke perdorur strukturen sockaddr\_in pasuar nga funksioni accept().Eshte shume e thjeshte:

|  |  |
| --- | --- |
| 12 | char \*client\_ip = inet\_ntoa(client.sin\_addr);int client\_port = ntohs(client.sin\_port); |

Ne pranuam nje lidhje hyrese por u mbyll menjehere.Kjo nuk eshte produktive.Jane shume gjera te ndryshme qe mund te behen nga lidhja ne rrjet qe kur themelohet.Megjithese lidhja eshte themeluar per qellimin e komunikimit.Keshtu le ti kthejme nje pergjigje klientit:

Ne me shume thjeshtesi mund te perdorim funksionin write() per te shkruar dicka ne nje soket per lidhjet hyrese dhe klienti do ta shikoje ate.Me poshte jepet nje shembull:

|  |  |
| --- | --- |
| 1234567891011121314151617181920212223242526272829303132333435363738394041424344454647484950515253 | #include<stdio.h>#include<string.h>    //strlen#include<sys/socket.h>#include<arpa/inet.h> //inet\_addr#include<unistd.h>    //write int main(int argc , char \*argv[]){    int socket\_desc , new\_socket , c;    struct sockaddr\_in server , client;    char \*message;         //Krijimi I soketit    socket\_desc = socket(AF\_INET , SOCK\_STREAM , 0);    if (socket\_desc == -1)    {        printf("Soketi nuk mund te krijohet");    }         //Pergatit strukturen sockaddr\_in    server.sin\_family = AF\_INET;    server.sin\_addr.s\_addr = INADDR\_ANY;    server.sin\_port = htons( 8888 );         //Lidhjet    if( bind(socket\_desc,(struct sockaddr \*)&server , sizeof(server)) < 0)    {        puts("Lidhja deshtoi");        return 1;    }    puts("Lidhja u krye");         //Degjo    listen(socket\_desc , 3);         //Prano lidhjet hyrese    puts("Waiting for incoming connections...");    c = sizeof(struct sockaddr\_in);    new\_socket = accept(socket\_desc, (struct sockaddr \*)&client, (socklen\_t\*)&c);    if (new\_socket<0)    {        perror("Aplikimi deshtoi");        return 1;    }         puts("Lidhja u aplikua");         //Pergjigja qe I behet klientit    message = "Hello Client\n";    write(new\_socket , message , strlen(message));         return 0;} |

Nis kodin me lart ne nje terminal.Dhe me pas lidhu ne kte server duket perdorur klientin telnet nga nje tjeter terminal dhe ti duhet te shohesh

$ telnet localhost 8888

Trying 127.0.0.1...

Connected to localhost.

Escape character is '^]'.

Hello Client

Connection closed by foreign host.

Keshtu klienti(telnet) mori nje pergjigje nga serveri.

Ne mund te shohim se lidhja u mbyll menjehere mbas ketij mesazhi sepse serveri mbyllet mbasi pranon dhe dergon mesazhin.Nje server si [www.google.com](http://www.google.com) eshte gjithmone I hapur per lidhjet hyrese

Keshtu ne nenkuptojm se serveri eshte supozuar te vazhdoj gjate gjithe kohes qe ka lidhje me internetin.Keshtu ne duhet te mbajme serverin tone te hapur non-stop.Menyra me e thjeshte per ta bere kete eshte te fusim funksionin accept() ne nje lak se keshtu ai mund te marre pergjigje ne lidhjet hyrese ne real-time.

**Live Server**

Keshtu nje Server Live duhet te qendroje gjithmone I tille.Le te shohim kodin me poshte:

|  |  |
| --- | --- |
| 12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455 | #include<stdio.h>#include<string.h>    //strlen#include<sys/socket.h>#include<arpa/inet.h> //inet\_addr#include<unistd.h>    //write int main(int argc , char \*argv[]){    int socket\_desc , new\_socket , c;    struct sockaddr\_in server , client;    char \*message;         //Krijimi I soketit    socket\_desc = socket(AF\_INET , SOCK\_STREAM , 0);    if (socket\_desc == -1)    {        printf("Soketi nuk mund te krijohet");    }         //Pergatit strukturen sockaddr\_in    server.sin\_family = AF\_INET;    server.sin\_addr.s\_addr = INADDR\_ANY;    server.sin\_port = htons( 8888 );         //Ngurtesim    if( bind(socket\_desc,(struct sockaddr \*)&server , sizeof(server)) < 0)    {        puts("Lidhja deshtoi");        return 1;    }    puts("Lidhja u krye");         //Degjo    listen(socket\_desc , 3);         //Aplikimi I lidhjeve hyrese    puts("Waiting for incoming connections...");    c = sizeof(struct sockaddr\_in);while( (new\_socket = accept(socket\_desc, (struct sockaddr \*)&client,(socklen\_t\*)&c)))    {        puts("Lidhja u pranua");                 //Pergjigja e klientit        message = "Hello Client \n";        write(new\_socket , message , strlen(message));    }         if (new\_socket<0)    {        perror("Aplikimi deshtoi");        return 1;    }         return 0;} |

Ne nuk kemi penfunduar ketu.Vetem akseptimi u fut ne nje lak.

Tani nise programin ne 1 terminal, dhe hap 3 terminal te ndryshme.Nga cdo keto 3 terminalet lidhu me serverin ne portin e caktuar.

Cdo terminal duhet te shfaqe si me poshte:

$ telnet localhost 8888

Trying 127.0.0.1...

Connected to localhost.

Escape character is '^]'.

Hello Client

The serveri duhet te shfaqe :

Lidhja u krye.

Duke pritur per lidhje te tjera...

Lidhja u aplikua

Lidhja u aplikua

Lidhja u aplikua

Keshtu tani serveri eshte duke vazhduar nonstop dhe terminalet me klientin telnet jane lidhur gjithashtu nonstop.Te gjitha teminalet me telnet do te shfaqin "Connection closed by foreign host."
Shume mir deri tani.Por akoma nuk eshte nje komunikim efektiv ndermjet serverit dhe klientit.Serveri pranon lidhjet ne nje lak dhe me pas I dergon atyre nje mesazh, me pas ai nuk ben asgje me to.Gjithashtu ai nuk eshte ne gjendje te mbaje me shume se 1 lidhje per kohe.

Keshtu tani eshte koha te krijojm nje server qe mund te mbaje shume lidhje ne te njejten kohe.

**Mbajtja e shume lidhjeve ne server me “threads”**

Per te mbajtur cdo lidhje ne duhet te ndajme kodin qe te veproje per kohe me serverin kryesor duke pranuar cdo lidhje te autorizuar.

 Nje menyre eshte duke I ruajtur keto duke perdorur temat ose “threads”.Serveri kryesor pranon lidhjet dhe krijon nje teme te re per te mbajtur komunikim per lidhjen e caktuar, dhe me pas serveri kthehet per te pranuar lidhje te reja
Ne Linux temat mund te kryhen me librarine pthread(temat posix).Do te jete shume mire nqs lexoni per kete librarine ne tutoriale te vogla nqs nuk di asgje rreth tyre.Megjithate perdorimi nuk eshte shume I komplikuar.

Tani ne duhet te perdorim temat per te krijuar mbajtese per cdo lidhje qe serveri do te pranoje.Shembulli me poshte I koduar ne gjuhen C per krijim e serverit me temat:

|  |  |
| --- | --- |
| 1234567891011121314151617181920212223242526272829303132333435363738394041424344454647484950515253545556575859606162636465666768697071727374757677787980818283 | #include<stdio.h>#include<string.h>    //strlen#include<stdlib.h>    //strlen#include<sys/socket.h>#include<arpa/inet.h> //inet\_addr#include<unistd.h>    //write #include<pthread.h> //for threading , link with lpthread void \*connection\_handler(void \*); int main(int argc , char \*argv[]){    int socket\_desc , new\_socket , c , \*new\_sock;    struct sockaddr\_in server , client;    char \*message;         //Krijimi I soketit    socket\_desc = socket(AF\_INET , SOCK\_STREAM , 0);    if (socket\_desc == -1)    {        printf("Soketi nuk mund te krijohet");    }         //Pergatitja e struktures sockaddr\_in    server.sin\_family = AF\_INET;    server.sin\_addr.s\_addr = INADDR\_ANY;    server.sin\_port = htons( 8888 );         //Lidhja ose ngurtesimi    if( bind(socket\_desc,(struct sockaddr \*)&server , sizeof(server)) < 0)    {        puts("Lidhja deshtoi");        return 1;    }    puts("Lidhja u krye");         //Degjo    listen(socket\_desc , 3);         //Pranimi I lidhjeve hyrese    puts("Duke pritur per lidhjet...");    c = sizeof(struct sockaddr\_in);while( (new\_socket = accept(socket\_desc, (struct sockaddr \*)&client, (socklen\_t\*)&c)))    {        puts("Lidhja u pranua");                 //Dergimi I mesazhit tek klienti        message = "Hello Client\n";        write(new\_socket , message , strlen(message));                 pthread\_t sniffer\_thread;        new\_sock = malloc(1);        \*new\_sock = new\_socket;        if( pthread\_create( &sniffer\_thread , NULL ,  connection\_handler , (void\*) new\_sock) < 0)        {            perror("nuk munda te krijoj temen");            return 1;        }                 //pthread\_join( sniffer\_thread , NULL);        puts("Mbajtesi u caktua");    }         if (new\_socket<0)    {        perror("Aplikimi deshtoi");        return 1;    }         return 0;} /\* \* Kjo do te mbaje lidhjet per cdo klient te lidhur \* \*/void \*connection\_handler(void \*socket\_desc){    //Merr pershkruesit e soketit    int sock = \*(int\*)socket\_desc;         char \*message;         //Dergo mesazhe tek klienti    message = "Pershendetje, une jam mbajtesi yt I lidhjeve\n";    write(sock , message , strlen(message));         message = "Eshte detyra ime per te kominikuar me ty";    write(sock , message , strlen(message));         //Liro pointerin e soket    free(socket\_desc);         return 0;} |

Nis serverin me lart dhe hap 3 terminal si me pare.Tani serveri do te krijoje tema per cdo klient qe do te lidhet me te

Terminalet me telnet do te shfaqin

$ telnet localhost 8888

Trying 127.0.0.1...

Connected to localhost.

Escape character is '^]'.

Hello Client

Pershendetje, une jam mbajtesi yt I lidhjeve

Eshte detyra ime per te komikuar me ty

Kjo duket shume mire, por mbajtesi I komunikimit eshte gjithashtu pak dembel.Mbas pershendetjes ai e mbyll programin.Ai duhet te qendroje I gjalle dhe te komunikoje me klientin.Nje menyre eshte duke e bere te mbaje lidhjen te prese per disa mesazhe nga klienti per sa gjate ai eshte I lidhure.Ne qofte se klienit shkeputet, mbajtesi I lidhje duhet te perfundoje.

Keshtu mbajtesi I lidhjes mund te rishkruhet si me poshte:

|  |  |
| --- | --- |
| 1234567891011121314151617181920212223242526272829303132 | /\* \* Kjo do te mbaje lidhje per cdo klient \* \*/void \*connection\_handler(void \*socket\_desc){    //Merr pershkruesin e soketit    int sock = \*(int\*)socket\_desc;    int read\_size;    char \*message , client\_message[2000];         //Dergo mesazhin tek klienti    message = " Pershendetje, une jam mbajtesi yt I lidhjeve n";    write(sock , message , strlen(message));         message = "Tani shtyp dicka dhe un duhet ta kthej mbrapsht tek ty\n”;    write(sock , message , strlen(message));         //Merr nje mesazh nga klienti    while( (read\_size = recv(sock , client\_message , 2000 , 0)) > 0 )    {        //Dergo mesazhin mbrapsht tek klienti        write(sock , client\_message , strlen(client\_message));    }         if(read\_size == 0)    {        puts("Klienti u shkeput");        fflush(stdout);    }    else if(read\_size == -1)    {        perror("Marrja deshtoi");    }             //Liro pointerin e soketit    free(socket\_desc);         return 0;} |

Mbajtesi I lidhjes me larte do te marre disa te dhena nga klienti dhe do ta ktheje ate mbrapsht te njejte.I thjeshte!.Tani me poshte jepet prodhimi I kodit

$ telnet localhost 8888

Trying 127.0.0.1...

Connected to localhost.

Escape character is '^]'.

Hello Client

Pershendetje, une jam mbajtesi yt I lidhjeve

Tani shtyp dicka dhe un duhet ta kthej mbrapsht tek ty Hello

Hello

Si jeni ju

Si jeni ju

Une jam shume mire

Une jam shume mireKeshtu tani ne kemi nje server komunikues.Tani eshte shume ndihmues.**Lidhja me librarin pthread**

**Lidhja me librarine pthread**

Kur kompilojme programet qe perdorin librarine pthread ne duhet ta lidhim me kete librari.Me poshte jepet menyra se si mund ta kryejme lidhjen.

$ gcc program.c -lpthread

**Konkluzioni**

Deri tani ju duhet te keni mesuar gjerat me te thjeshta ne programimit te soketave ne C. Ti mund te perpiqesh to besh disa eksperimente si te shkruash nje program qe vepron si nje chat ose dicka te ngjashme me kete.