

بسم الله الرحمن الرحيم

برنامه نویسی تحت شبکه socket programming

عنوان مقاله : Socket Programing with C++

کلمات کلیدی:

IP : در شبکه هر کامپیوتر دارای آدرس میباشد , این عدد آدرس ۳۲ بیتی , به عبارتی ۴ قسمتی را IP گویند .

TCP : یکی از پروتکل های لایه انتقال در شبکه اتصالاتگر میباشد (Transmission Control Protocol) .

UDP : یکی از پروتکل های لایه انتقال در شبکه غیراتصالاتگر میباشد.

CLIENT : کامپیوتری که در شبکه سرویس گیرنده است.

SERVER : کامپیوتری که به درخواستهای سرویس گیرنده ها پاسخ میدهد.

خلاصه:

امروزه با گسترش ارتباطات و انواع شبکه های کامپیوتری جهت انجام امور نیاز مبرم به برنامه های تحت شبکه از جمله socket programming احساس میشود , لذا در این مقاله به ارائه مطالبی در این زمینه و ایجاد یک برنامه تحت شبکه (برنامه chat) میپردازم. در حقیقت این مقاله نقطه آغازی برای همه برنامه نویسانی خواهد بود که به نحوی مجبور خواهند شد برنامه کار بردی تحت شبکه اینترنت بنویسند.

برای یک ارتباط نیاز به موارد زیر است:

- ارتباط ادامه دار
- زبان مشترک
- معنای آدرس دهی

ارتباط عمومی:

پروتکلها : زبانهای مشترک اینترنت هستند.

انواع آدرس در اینترنت:

1- آدرس IP از جنس آدرس فیزیکی ۱۶۷, ۱۸۸, ۱۲, ۱۶۱

2- نام Domain از جنس آدرس منطقی ex.iraji.com

آدرس IP ۳۲ بیتی است که به ۴ قسمت تقسیم می گردد محدوده اعداد در هر قسمت از ۰-۲۵۵ می باشد.

5 طبقه بندی برای آدرس IP وجود دارد :

توابع API در C :

GETHOST BY NAME – NAME IP ADDRESS

GET HOST BY ADDRESS – IP ADDRESS AQME

برنامه های کاربردی مشترک:

در هر مورد پروتکل نمونه تعریف شده است.

شماره پورت:

یک عدد 16 بیت صحیح است . برای ارتباط نیاز به آدرس IP و شماره پورت می باشد.

در unix 1024 پورت اول برای هر دو نوع پورتکلها است و web known ports نامیده می شود .

مدل tcp مبتنی بر اتصال است
قابل اطمینان ftp, telnet, http, smtr
رشته ای است مسیل و نهر (transmission control protocol / internet protocol)

مدل udp غیر قابل اتصال
غیر قابل اطمینان tftp, nfs
دیتا گرامی است. بر اساس packet (user datagram protocol) چکسام

مفهوم socket :

فرض کنید در محیطی کار میکنید که فقط با فایل ها سر و کار داریم, در چنین سیستمی ورودی خروجی هم (I/O) میتواند توسط سیستم فایلها مدیریت شود. به عنوان مثال, صفحه نمایش را همانند یک فایل متنی در نظر بگیرید برای اینکه اطلاعات را در صفحه نمایش بفرستیم باید فایل را از نوع نوشتنی باز کنیم و اگر محتوای اشاره گر فایل خالی نبود یعنی سیستم عامل به فایل حافظه اختصاص داد میتوان در فایل اطلاعات را نوشت.

میتوانیم این ساختار را برای ارتباط دو برنامه کاربردی روی دو کامپیوتر گسترش دهیم, یعنی باید یک ارتباط با یک کامپیوتری شبکه) با یک آدرس Ip خاص (و برنامه ای خاص روی کامپیوتر با آدرس پورت مشخص بگشاییم.

سوکت, ترکیبی از یک آدرس IP و پورت TCP و یا پورت UDP است. یک برنامه, سوکتی را با مشخص نمودن آدرس IP مربوط به کامپیوتر و نوع سرویس TCP (برای تضمین توزیع اطلاعات و یا UDP و پورتهای که نشاندهنده برنامه است, مشخص می نماید. آدرس IP موجود در سوکت, امکان آدرس دهی کامپیوتر مقصد را فراهم و پورت مربوطه, برنامه ای را که داده ها برای آن ارسال می گردد را مشخص می نماید. در زیر نام چند پروتکل برنامه کاربردی و آدرس پورت آنها مشخص شده است:

بسیاری از برنامه های کاربردی تحت شبکه از یک شماره پورت خاصی برای ارسال و دریافت اطلاعات استفاده میکنند. به عنوان مثال برنامه فرستادن ایمیل به پورت ۲۵ گوش میکند, ما در برنامه های کاربردی خود می توانیم از پورت دلخواه استفاده کنیم, به عنوان مثال شماره سوکت در یک برنامه تحت شبکه میتواند ۲۰۰۰: ۱۲۷,۶۷,۷۸,۹۷ باشد که قالب کلی این آدرس به صورت Ip:Port میباشد.

مفهوم Sap :

در سمت client & server بسته های شکسته شده برنامه های کاربردی مربوطه با عبور از پورت مربوطه از لایه های دیگر شبکه عبور میکنند. این شماره پورت واسط بین لایه برنامه کاربردی و انتقال در حقیقت همان Sap میباشد. به شکل زیر نگاه کنید.

بنابراین ارتباط بین دو برنامه تحت شبکه مبادله اطلاعات بین دو نقطه TSap روی دو کامپیوتر در شبکه میباشد. همانطور که میتوان چندین فایل را همزمان باز نگه داشت در یک برنامه تحت شبکه همزمان میتوان چندین ارتباط باز و فعال داشته باشیم. مثلاً از برنامه Yahoo messenger استفاده میکنیم در حقیقت ما از برنامه سمت client آن استفاده میکنیم و در server yahoo برنامه ای به پورتهای که برنامه client اطلاعات را به آن میفرستد گوش میدهد.

انواع سوکت:

سوکت ها به طور کلی به ۲ نوع میباشند:

1 - سوکت های نوع stream که از نوع اتصالگرا هستند, یعنی باید قبل از مبادله اطلاعات یک اتصال به صورت لوله فرضی بین client و server بوجود میاید, این نوع سوکت ها بر مبنای tcp هستند.

2 - سوکت های datagram که از نوع بدون اتصال هستند ، یعنی نیاز به برقراری ارتباط فرضی ندارند ، پس هیچ امنیتی در این نوع ارتباط وجود ندارد، این نوع سوکت ها بر مبنای udp هستند. همواره باید به این نکته توجه کرد که در برنامه های تحت شبکه ارتباطات دو طرفه میباشد، یک برنامه ای که در سمت سرویس گیرنده اجرا میشود و دیگری در سمت سرویس دهنده. برنامه نویسی سوکت مربوط به لایه کاربردی میباشد و انتقال بسته های اطلاعاتی به لایه های دیگر بر عهده سیستم عامل میباشد.

قبل از وارد شدن در مقوله برنامه نویسی سوکت بد نیست الگوریتم کل کاری که بایستی در سمت سرویس دهنده همچنین در سمت مشتری انجام شود را بررسی نماییم:

الف) یک سوکت را که مشخصه یک ارتباط است بوجود بیاوریم . این قسمت باید نوع ارتباط خود با سیستم عامل را (tcp) یا (udp) معرفی نماییم . این کار در محیط یونیکس توسط تابع socket() انجام می گیرد.

ب) به سوکتی که باز کرده اید آدرس پورت نسبت می دهیم بدین صورت تمام بسته های Tcp یا upd ای که آدرس پورت مقصدشان با شماره پورت برنامه شما مطابقت دارد به سمت برنامه شما فرستاده می شود این کار توسط تابع سیستمی bind() انجام می شود. به این پورت ، پورت منفعل می گویند.

ج) حال باید به سیستم عامل بگوییم کارش را برای پذیرش تقاضاهای ارتباط شروع نماید. این کار توسط تابع سیستمی listen() انجام می شود . چون ممکن است تعداد تقاضاهای ارتباط زیاد باشد باید حد اکثر تعداد ارتباط tcp را که می توان پذیرفت مشخص کرد و برای آن بافر در نظر گرفت.

د) در پایان باید از سیستم عامل بخواهیم که یکی از ارتباطات معلق را به برنامه شما معرفی کند . این کار توسط تابع accept() انجام می گیرد.

ه) از دستورات send() و recv() برای مبادله داده استفاده می شود.
و) در پایان باید ارتباط خاتمه یابد . به دو صورت است:

در برنامه مشتری بایستی اعمال زیر را انجام داد:

الف) یک سوکت بوجود بیاوریم
ب) نیازی نیست مانند سرویس دهنده به سوکت خود آدرس پورت نسبت دهیم یعنی لزومی به استفاده از دستور bind نیست چرا که برنامه سمت client منتظر تقاضای دیگران نیست بلکه خودش متقاضی برقراری ارتباط است . بنا بر این با تابع connect تقاضا را به سمت server می فرستیم.
ج) از توابع send() یا recv() برای ارسال و دریافت داده اقدام کنید.
د) ارتباط را با توابع close یا shut down می بندیم.

توابع مورد استفاده در socket programming در C :

* توابع socket غیر واقعی :

Ret = parse_net work_args (args & argv & host & port & errmes)

تنظیمات مربوط به شبکه را انجام می دهد و ویرایش می کند. به دنبال host و port می گردد و آنها را منتقل می کند . در صورت موفقیت صفررا برمی گرداند.

** قسمت server :

1) - Port_sk = socket(af_inet ,sock_stream):

2) - Port_sk = tcp_passive_open (port)

در قسمت server اجرا مي شود.
Port = تعداد رورت هايي كه به كار مي رود .
Port- sk .توصيفگر socket است

3) - Bind (port_sk, &bind_addr , addr_len);

4) - Client_sk = accept (port_sk , &bind_addr ,&len);

*** قسمت client :

1)-Server_sk = tcp_active_open (host ,port)

Host = نام ماشين server
Port = شماره پورتي كه به كار مي رود
Server-sk = توصيفگر socket است.
آدرس IP و HOST پيدا مي شود:

Host Ip = gethost by name(host name);

2) Server_sk = secket (af_inet و sock_stream;);

3) connect(server_sk و &bind_addr و addr_len);

سوكت Server_sk مي تواند براي صحبت با سرور به كار رود.
Header files socket هاي اصلي در زبان c براي برنامه نويسي

<sys/ types.h>.لازمه انواع داده اي است

<errno.h>.ها مي باشد error شماره

<sys/sochet.n>.و سيستمي را شامل مي باشد socke ساختارها و ثابت هاي

<netdb.h.h>.ساختارهاي مربوطه به شبكه اي را شامل مي باشد

<netinet/in.h >.را شامل مي شودsockets توابع مربوط <arpa/inet.h>. توابع کاربردي را شامل مي باشد

توابع مديريت داده در شبكه :

- * get host by name - IP و نام dns يك host را بر مي گردان .
- * get server by name - نام service و پروتكل را بر مي گرداند .
- * get host name - نام host را در شبكه محلي بر مي گرداند.

* get server by name get serv by port get servent

* getproto by name get proto by number get protobyeat

* get net by name get net by add get net ent

مقايسه ارسال و دريافت داده ها به روش udp :

روال در هر دو برنامه سمت client ، server به يك شكل است:

الف) يك سوكت از نوع ديتا گرام ايجاد كنيد . اين كار با فراخواني تابع socket () با پارامتر sock-dgram انجام مي شود

ب) با تابع bind () به سوكت ايجاد شده آدرس پورت مورد نظرتان را نسبت دهيد.

ج) حال مي توانيد منتظر دريافت و يا ارسال داده ها باشيد (در دريافت آدرس برنامه مبداء (آدرس ip و port مشخص مي گردد و ارسال امكان پذير است).

ارسال و دريافت با توابع انجام مي شود.

د) سوكت ايجاد شده را ببنديد.

نتيجه گيري:

بر اساس تمام برنامه های کاربردی تحت شبکه لایه چهارم مفهومی به نام سوکت است . این مفهوم به منظور برقراری ارتباط برنامه های تحت شبکه و تبادل جریان داده ها بین پروسه ها ابداع شد . لذا این مقاله نقطه آغازی برای همه برنامه نویسانی خواهد بود که به نحوی مجبور خواهند شد برنامه کاربردی تحت شبکه اینترنت بنویسند.

منابع:

- * Internet working with tcp/ ip : uelumes I II III Douglas comer , prentice hall , 2000
- * the whole internet user s guide & catalog by ed kroll; oreilly & associates
- * socket Api programmer s refrence 2003

نویسنده : مهندس محمد صا بر ایرجی (shantia20042004)
ایمیل : Shantia20042004@yahoo.com

کلیه حقوق این مقاله متعلق به نویسنده و سایت برنامه نویسی می باشد .
استفاده از مطالب این مقاله در صورت ذکر منبع مجاز است .

www.barnamenevis.org