دانشگاه جامع علمی و کاربردی شبستر

جزوه درسی برای درس:

برنامه نویسی شبکه - Socket Programing

گردآوری : ج . پ

بهار 87

مروری بر مفاهیم شبکه :

عاملی که تمام شبکه های مختلف را به صورت موفقیت آمیز به هم پیوند زده است ، تبعیت همه آنها از مجموعه پروتکلی است که تحت عنوان TCP/IP دردنیا شناخته شده است . دقت کنید که عبارت خلاصه شده TCP/IP میتواند به دو موضوع مختلف اشاره داشته باشد :

• مدلTCP/IP : این مدل یک ساختار چهار لایه ای برای ارتباطات گسترده تعریف می نماید که آن را در ادامه بررسی می کنیم.

• پشته پروتوکلهایTCP/IP (TCP/IP Protocol Stack) :

مجموعه ای شامل بیش از صد پروتکل متفاوت است که برای سازماندهی کلیه اجزاء شبکه اینترنت بکار میرود. در این دوره تعدادی از این پروتکل ها را که عمومیت و استفاده فراگیر دارند معرفی خواهیم کرد. به خاطرداشته باشیدکه دراین دوره جزئیات TCP/IP رابه شماآموزش نخواهیم داد.

TCP/IP بهترین پروتکل شبکه بندی دنیا نیست! پروتکلهای بهینه تر از آن نیز وجود دارند؛ و لیکن فراگیرترین و محبوبترین تکنولوژی شبکه بندی در دنیای کامپیوتر محصوب می شود. شاید بزرگترین حسن TCP/IP آنباشدکه بدون پیچیدگی زیاد، به خوبی کار میکند! اینترنت بر اساس TCP/IP بنا شده و بیشتر حملات نیز مبتنی بر مجموعه پروتکلهایTCP/IP هستند.

طراحی شبکه ها و اصول لایه بندی

برای طراحییک شبکهٔ کامپیوتری، مسائل و مشکلات بسیار گسترده ومتنوعی وجود دارد که باید به نحوی حل شود تا بتوان یک ارتباط مطمئن و قابل اعتماد بین دو ماشین در شبکه برقرار کرد.

این مساول و مشکلات همگی از یک سنخ نیستند و منشاء و راه حل مشابه نیز ندارند؛ بخشی از آنها توسط سخت افزار و بخش دیگر با تکنیکهای نرم افزاری قابل حل هستند. به عنوان مثال نیاز برای ارتباط بی سیم بین چند ایستگاه در شبکه ، طراح شبکه را مجبور به استفاده از مدولاسیون آنالوگ در سخت فزار مخابراتی خواهد کرد ولی مسئله هماهنگی درارسال بسته هاازمبدأبه مقصدیا شماره گذاری بسته ها برای بازسازی پیام و اطمینان از رسیدن یک بسته ، با استفاده از تکنیکهای نرم افزاری قابل حل است.

بهمین دلیل برای طراحی شبکه های کامپیوتری، باید مسائل و مشکلاتی که برای برقراریک ارتباط مطمئن، ساده و شفاف بین دو ماشین در شبکه وجود دارد، دسته بندی شده و و راه حلهای استاندارد برای آنها ارائه می شود.

در زیر بخشی از مسائل طراحی شبکه ها عنوان شده است :

اولـین موضـوع چگـونگی ارسـال و دریافـت بیتهـای اطلاعـات بـه صـورت یـک سـیگنال الکتریکـی ، الکترومغناطیسی یا نوری است، بسته به اینکه آیا کانال انتقال سیم مسی ، فیبر نوری، کانال ماهواره ایبا خطوط مایکروویو است. بنابراین تبدیل بیتها به یک سیگنال متناسب با کانال انتقال یکی از مسائل اولیه شبکه به شمار میرود.

مسئله دوم ماهیت انتقال است که می تواند به یکی از سه صورت زیر باشد:

Simplex: ارتباط یک طرفه (یک طرف همیشه گیرنده و طرف دیگر همیشه فرستنده می باشد)

Half Duplex: ارتباط دوطرفه غیر همزمان (هر دو ماشین هم می توانند فرستنده یا گیرنده باشند ولی نـه به صورت همزمان، بلکه یکی از طرفین ابتدا ارسال می کند ، سپس ساکت مس شود تا طرف مقابل ارسال داشته باشد)

FullDuplex: ارتباط دوطرفه همزمان (مانند خطوط مایکروویو)

مسئله سوم مسئله خطاووجودنویز روی کانالهای ارتباطی است بدین معنا که ممکن است در حین ارسال داده ها بر روی کانال فیزیکی تعدادی از بیتها دچار خرابی شود؛ چنین وضعیتی که قابل اجتناب نیست باید تشخیص داده شده و داده های فاقد اعتبار دور ریخته شودو مبداء آنها را از نو ارسال کند.

باتوجه به اینکه در شبکه ها ممکن است مسیرهای گوناگونی بین مبداء و مقصد وجود داشته باشد؛ بنـابراین پیدا کردن بهترین مسیر و هدایت بسته ها، از مسائل طراحی شبکه محصوب می شود. در ضـمن ممکـن اسـت یـک پیـام بزرگ به واحدهای کوچکتری تقسیم شده و از مسیرهای مختلفی به مقصد برسد. بنابر این بازسازی پیام از دیگر مسـائل شبکه به شمار می آید.

ممکن است گیرنده به دلایلی نتواند باسرعتی که فرستنده بسته هاییک پیام را ارسال می کند آنها را دریافت کند، بنابراین طراحی مکانیزم های حفظ هماهنگی بین مبدأ و مقصد از دیگر مسائل شبکه است.

چون ماشین های فرستنده وگیرنده متعددی در یک شبکه وجود دارد مسائلی مثل ازدحام ، تداخل و تصادم در شبکه ها بوجود می آید که این مشکلات به همراه دیگر مشکلات باید در سخت افزار و نرم افزار شبکه حل شود

طراح یک شبکه باید تمام مسائل شبکه را تجزیه و تحلیل کرده و برای آنها راه حل ارائه کند ولی چون این مسائل دارای ماهیتی متفاوت از یکدیگر هستند، بنابراین طراحییک شبکه باید بصورت "لایه به لایه" انجام شود . به عنوان مثـال وقتی قرار است یک شبکه به گونه ای طراحی شود که استگاه ها بتوانند انتقال فایل داشته باشند، اولین مسئله ای که باید به آن بیندیشند طراحییک سخت افزار مخابراتی برای ارسال و دریافت بیتها رو کانال فیزیکی اسـت. اگـر چنـین سـخت افزاری طراحی شود ، می تواند بـر اسـاس آن اقـدام بـه حـل مسـئله خطاهـای احتمـالی در داده هـا نمایـد؛ یعنـی زمـانی مکانیزمهای کنترل و کشف خطا مطرح می شود که قبل از آن سخت افزار ومخابره داده ها طراحـی شـده باشـد . بعـد از

این دو مرحله طراحی باید مکانیزمهای بسته بندی اطلاعات ، آدرس دهی ماشین ها و مسیریابی بسته ها طراحی شود. سپس برای بقیه مسائل نظیر آدرس دهی پروسه ها و چگونگی انتقال فایل راه حل ارائه شود.

طراحی لایه ای شبکه به منظور تفکیک مسائلی است که باید توسط طراح حل شود و مبتنی بر اصول زیر است :

هرلایه وظیفه مشخصی دارد و طراح شبکه باید آنها را به دقت تشریح کند.

هرگاه سرویسهایی که باید ارائه شود از نظر ماهیتی متفاوت باشند ، باید لایه به لایه و جداگانه طراحی شود.

وظیفه هر لایه نباید انقدر زیاد باشد که تمایز لایه ها از دیدگاه سرویسهای ارائه شده نا مشخص باشد و آنقدر کم باشد ، که وظیفه و خدمات یک لایه ، پیچیده و نا مشخص شود.

درهرلایه جزئیات لایه های زیرین نادیده گرفته می شود و لایه های بالایی باید در یک روال ساده و ماجولار از خدمات لایه زیرین خود استفاده کنند.

باید مرزهای هر لایه به گونه ای انتخاب شود که جریان اطلاعات بین لایه ها ، حداقل باشد.

برای آنکه طراحی شبکه ها سلیقه ای و پیچیده نشود سازمان جهانی استاندارد (ISO) ، مدلی هفت لایه ای برای شبکه ارائه کرد ، به گونه ای که وظائف و خدمات شبکه در هفت لایه مجزا تعریف و ارائه می شود . این مدل هفت لایه ای ، OSI نام گرفت. هر چند در شبکه اینترنت از این مدل استفاده نمی شود و بجای آن از مدل چهار لایه ای به نام TCP/IP تعریف شده است ، ولیکن بررسی مدل هفت لایه ایOSI ، به دلیل دقتی که در تفکیک و تبیین مسائل شبکه در آن وجود دارد ، با ارزش خواهد بود.

مدل چهار لایه ایTCP/IP

همانگونه که اشاره شداین مدل یک ساختار چهار لایه ای برای شبکه عرضه کرده است. شکل زیر این مدل را به تصویر کشیده است. اگر بخواهیم این مدل چهار لایه ای را با مدل OSI مقایسه کنیم ، لایه اول از TCP/IPیعنی لایه دسترسی به شبکه تلفیقی از وظائف لایه فیزیکی ، لایه پیوند داده از مدل OSI خواهد بود .لایه دوم از TCP/IP معادل لایه سوم از مدل OSIیعنی لایه شبکه است. لایه سوم از مدل TCP/IP همنام و معادل با لایه چهارم از مدل OSIیعنی لایه انتقال خواهد بود. لایه پنجم (جلسه) و لایه ششم(ارائه) از مدل OSI در مدل TCP/IP وجودندارندووظائف آنها درصورت لزوم درلایه چهارم از مدل TCP/IP ادغام شده است . لایه هفتم از مدل OSI معدل بخشی از لایه چهارم از مدل TCP/IP است. در شکل زیر دو مدل TCP/IP و OSI باهم مقایسه شده اند .

در ادامه چهارلایه مدل TCP/IP را با هم بررسی خواهیم کرد .

لایه اول از مدل TCP/IP : لایه واسط شبکه

دراین لایه استانداردهای سخت افزار ، نرم افزار راه انداز (Device Driver) و پروتکلهای شبکه تعریف می شود. این لایه درگیر با مسائل فیزیکی ، الکتریکی و مخابراتی کانال انتقال ، نوع کارت شبکه و راه اندازهای لازم برای نصب کارت شبکه می باشد. در شبکه اینترنت که می تواند مجموعه ای از عناصر غیر همگن و نا مشابه را به هم پیوند بزند انعطاف لازم در این لایه برای شبکه های گوناگون و ماشینهای میزبان فراهم شده است. یعنی الزام ویژه ای در بکارگیری سخت افزار ارتابطی خاص ، در این لایه وجود ندارد. ایستگاهی که تصمیم دارد به اینترنت متصل شود بایستی با استفاده از پروتکل های متعدد و معتبر و نرم افزار ره انداز مناسب ، بنحوی داده ه های خودش را به شبکه تزریق کند. بنابراین اصرار و اجبار خاصی در استفاده از یک استاندارد خاص در این لایه وجود ندارد. تمام پروتکلهای LAN و MAN دراین لایه قابل استفاده اند.

یک ماشین میزبان می تواند از طریق شبکه محلی ، فریمهای اطلاعاتی را به زیر شبکه تزریق کند به این نحو که بسته های راه دور(Distance Packet) را که مقصدشان خارج از شبکه محلی است ، به مسریاب از پیش تعریف شده ، هدایت نماید. شبکه های محلی از طریق یک یا چند مسیریاب می توانند به اینترنت متصل می شوند. بنابراین یک بسته اطلاعاتی که از لایه بالاتر جهت ارسال به یک مقصد ، به لایه اول در مدل TCP/IP تحویل می شود.، نهایتاً در قسمت "فیلد داده (Payload/Data Field) " از فریم شبکه محلی قرار می گیرد و مسیر خود را آغاز می نماید؛ پروتکلهائی که در لایه اول از مدل TCP/IP تعریف می شوند ، میتوانند مبتنی بر ارسال رشته بیت(در اینجا کوچکترین واحد اطلاعاتی که می تواند ارسال شود یک بیت خواهد بود Bit Oriented) یا مبتنی بر ارسال رشته بایت(در اینجا کوچکترین واحد اطلاعات که می تواند ارسال شود بایت خواهد بود Byte Oriented) باشند.

لایه دوم از مدل TCP/IP : لایه شبکه

این لایه در ساده ترین عبارت وظیفه دارد بسته های اطلاعاتی را که از این به بعد آنها را بسته هایIP می نامیم، روی شبکه هدایت کرده و از مبدأ تا مقصد به پیش ببرد. در این لایه چندین پروتکل در کنار هم وظیفه مسیریابی و تحویل بسته های اطلاعاتی از مبدأ تا مقصد را انجام میدهند. کلیدی ترین پروتکل در این لایه ، پروتکل IP نام دارد. برخی از پروتکل های مهم که یکسری وظائف جانبی برعهده دارند عبارتند از : RARP،ARP ، BOOTP،IGMP،ICMP،RIPو... این پروتکل ها را به اختصار تئضیح خواهیم داد ولی بیشترین تلاش ما در کالبدشناسی پروتکل IP خواهد بود.

دراین لایه یک واحد اطلاعاتی که بایستی تحویل مقصد شوند ، دیتاگرام نامیده می شوند. پروتکل IP می تواند یک دیتاگرام را در قالب بسته های کوچکتری قطعه قطعه کرده و پس از اضافه کردن اطلاعات لازم برای بازسازی ، آنها را روی شبکه ارسال می کند.

لازم است بدانید که در این لایه برای برقراری ارتباط بین مبدأ و مقصد بروش "بدون اتصال" خواهد بود و ارسال یک بسته IP روی شبکه ، عبور از مسیر خاصی را تضمین نمی کند. یعنی اگر دو بسته متوالی برایبک مقصد یکسان ارسال شود هیچ تضمینی در به ترتیب رسیدن آنها وجود ندارد، چون این دو بسته می توانند از مسیرهای متفاوتی به سمت مقصد حرکت نمایند. در ضمن در این لایه پس از آنکه بسته ای روییکی از کانالهای ارتباطی هدایت شد، از سالم رسیدن یا نرسیدن آن به مقصد هیچ اطلاعی به دست نخواهد آمد، چرا که در این لایه، برای بسته هایIP هیچگونه پیامی دریافت (NACK/ACK) بین عناصر واقع بر روی مسیر، ردوبدل نمی شوند؛ بنابراین سرویسی که در این لایه ارائه می شود نامطمئن است و اگر به سرویسهای مطمئن و اتصال گرا نیاز باشد در لایه بالاتر این نیاز تأمین خواهد شد.

دراین لایه مسیریابها بایستی از شرایط توپولوژی و ترافیکی شبکه اطلاعاتی را کس نمایتد تا مسیریابی به روش پویا انجام شود همچنین در این لایه باید اطلاعاتی درباره مشکلات یا خطاهای احتمالی در ساختار زیرشبکه بین مسیریابها و ماشینهای میزبان، مبادله شود. یکی دیگر از وظائف این لایه ویژگی ارسال "چند بخشی(Multicast)" است. یعنی یک ایستگاه قادر باشد به چندین مقصد گوناگون که در قالب یک گروه سازماندهی شده اند، بسته یا بسته هایی ارسال نماید.

لایه سوم از مدل TCP/IP: لایه انتقال

این لایه ارتباط ماشینهای انتهایی(ماشینهای میزبان) را در شبکه برقرار می کند یعنی می تواند بر اساس سرویسی که لایه دوم ارائه می کند یک ارتباط اتصالگرا و مطمئن(Reliable) ،برقرارنماید. این لایه برای عملیاتی نظیر ارسال صوت و تصویر که سرعت مهمتر از دقت و خطا است سرویسهایی سریع و نا مطمئن نیز فراهم شده است.

درسرویس مطمئنی که در این لایه ارائه می شود، مکانیزمی اتخاذ شده است که فرستنده از رسیدن و یا عدم رسید صحیح بسته به مقصد با خبر شود. در مورد سرویسهای مطمئن و نا مطمئن بعداً بحث خواهد شد. این لایه از یک طرف با لایه شبکه و از طرفی دیگر با لایه کاربرد در ارتباط است. داده های تحویلی به این لایه توسط برنامه های کاربردی و با صدا زدن توابع سیستمی تعریف شده در "واسط برنامه های کاربردی(Application Program Interface)" ارسال یا دریافت می شوند.

لایه چهارم از مدل TCP/IP : لایه کاربرد

دراین لایه بر اساس خدمات لایه های زیرین، سرویس سطح بالایی برای خلق برنامه های کاربردی ویژه و پیچیده ارائه می شود. این خدمات در قالب، پروتکاهای استانداردی همانند موارد زیر به کاربر ارائه می شود:

1. شبیه سازی ترمینال (Telnet/ Terminal Emulation)
2. انتقالفایل یا FTP

3. مدیریت پست الکترونیکی
4. خدمات انتقال صفحات ابرمتنی
5. ودهها‌پروتکل کاربردی دیگر ...

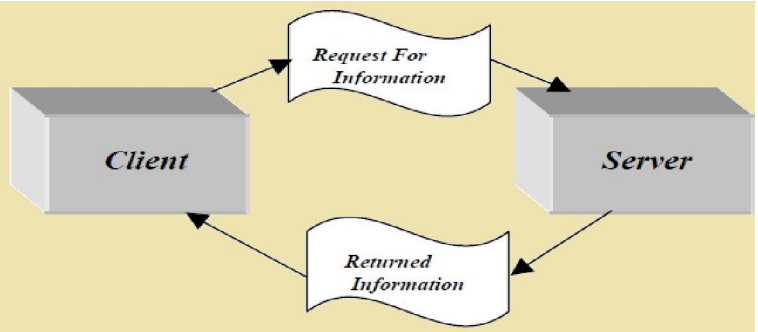
در پایان این قسمت باید خاطر نشان کنم که ارسال یک واحد اطلاعاتی از لایه چهارم پس از انجام پردازشهای لازم در لایه های زیرین به نحو مناسبی روی زیر شبکه تزریق شده و نهایتاً در ماشین مقصد، تحویل یک برنامه کاربردی خاص خواهد شد.

برنامه سازی شبکه و سوکت

سوکت به عنوان یک رابط و واسط میان برنامه و پروتوکل شبکهای مانندTCP ، مهمترین بخش از یک برنامه تحت شبکه محسوب می شود. برای درک بهترعملکرد سوکت، به بررسی مثالی می پردازیم. فرض کنید که یک برنامه نویس، برنامه ای رانوشته که مقادیری را در فایلی درون هارد دیسک ذخیره می کند. واقعیت امر آن است کهبرنامه نویس تنها از طریق متدهای موجود، اقدام به این عمل کرده و هیچ اطلاعاتی ازعملکرد فیزیکی پشت صحنه که سیستم عامل ندارد. بطور مشابه در مورد سوکت نیز، برنامهنویسان بدون توجه به جزئیات مربوط به کارت های شبکه نصب شده بر روی سیستم، نحوه ایجاد بسته های اطلاعاتی و موارد مشابه دیگر، با ایجاد سوکت و فراخوانی متدهایی ازآن عملیات مربوط به برقراری اتصال، ارسال، دریافت بسته های اطلاعاتی و پایان دادنبه ارتباط را انجام داده و اعمال سطح پایین تر، توسط سیستم عامل انجام میپذیرد.

مفهوم سرویس دهنده /مشتری

اگربرنامه ای راکه شروعکننده ارتباطاست،"برنامه مشتری"بنامیم قاعدتاًبرنامه ای که این ارتباط رامی پذیرد (ومنتظرآنبوده) "سرویس دهنده"نام خواهدگرفت.



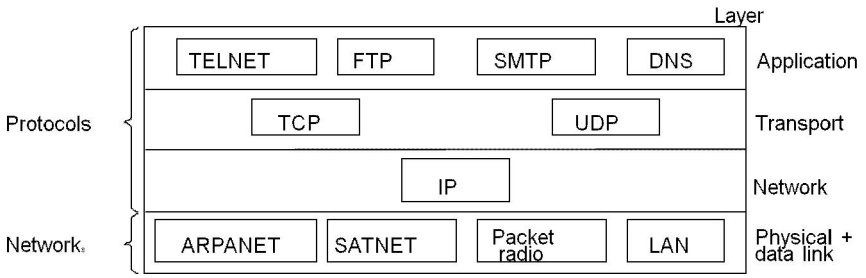
شکل (۱-۷) ارتباط بین سرویس دهنده و مشتری

تعریف عمومی: مشتری Client پروسه ای است که اصولاًنیازمندمقداری اطلاعات است و سرویس دهنده Server پروسه ای است که اطلاعاتی رادراختیاردارد و تمایل دارد تااین اطلاعات رابه اشتراک بگذاردومنتظرمیماندتایک متقاضی واحدی ازاین اطلاعات را طلب کندواوآنراتحویل بدهد.

انتخاب نوع سوکت ارتباطی

اساساً سوکت ها را به دو بخش اتصالگرا (Connection-Oriented) و بدون اتصال(Connection-Less)می توان تقسیم بندی کرد .

در سوکت های اتصال گرا مانندTCP ، برای مبادله داده ها بین دو ماشین، حتماًباید پیشتر، اتصالی برقرار شده باشد. اما در سوکت های بدون اتصال مانندUDP، نیازیبه برقراری اتصال نبوده و در عوض، به ازای هر بار ارسال داده به ماشین دیگر، آدرس IPماشین مقصد می بایست مشخص شود. از آنجایی که سوکت های اتصال گرا از محبوبیت بیشتریدر ایجاد برنامه های تحت شبکه برخوردار هستند، لذا در این بخش فقط به بررسیاین نوع سوکت ها پرداخته ایم .



در این بخش ، مراحلمختلف ایجاد برنامه های سرویس دهنده و سرویس گیرنده ای که ازTCP برای ارتباطاستفاده می کنند، را بررسی کرده و همچنین متدهای مختلف کلاس socketکه برای برقراریارتباط و همچنین مبادله داده ها وجود دارند، شرح خواهیم داد. برای شروع، ابتدابرنامه سرویس دهنده ساده ای ایجاد خواهیم کرد، سپس به بررسی برنامه سرویس گیرنده میپردازیم.

از آنجایی که نیاز به برنامه های تحت شبکه برای برقراری ارتباط کاربران با یکدیگر هر روز بیشتر می شود، لذا تأثیر بسزای سیستم های مربوطه، در بالا بردن بازده کاری، بر کسی پوشیده نیست. در این بخش، نوع خاصی از برنامه مورد توجه قرار نگرفته و مفاهیم کلی هر نرم افزار تحت شبکه، مورد بحث است.

استریم

همانطور که می دانید، روش های بسیاری برای ذخیره سازی اطلاعات در حافظه، فایل، ابزارهای ورودی/خروجی، خطوط ارتباط داخلی و خطوط ارتباطی شبکه وجود دارند. لازم به ذکر است که معمولاً داده ها به صورت بایت به بایت نوشته و خوانده می شوند که در مقابل ضریب اطمینان بالایی که این روش دارد، کارایی آن قابل توجه نیست. به همین جهت برای بالا بردن کارایی، می توان از روشی که استریم ها از آن استفاده می کنند، استفاده کرده و در هر لحظه، به

جای خواندن و نوشتن بایت به بایت، بر روی بلوکی از داده ها کار کرده و چندین بایت را مبادله نمود. در کار با فایل ها و برخی ابزارهای ورودی/خروجی و ارتباطات شبکه ای می توان از استریم ها برای بالا بردن سرعت عملکرد برنامه ها استفاده کرد. واضح است که برای ابزارهایی همچون CD-ROM ، استریم، عمل "نوشتن" را انجام نمی دهد. همچنین در خطوط ارتباطی شبکه نیز استریم ها از عمل جستجو، پشتیبانی نمی کنند.

خصوصیات

مهمترین خصوصیتTCP، اتصال گرا بودن آن و بدان معناست که تنها در صورتی دو کامپیوتر با یکدیگر ارتباط دارند که یک کانال اتصال با یکدیگر بر قرار کرده باشند (در بخش های بعدی با این خصوصیت، بیشتر آشنا خواهیدشد). در این روش امکان مبادله داده بدون برقراری اتصال وجود نداشته باشد، استریمی جهت اطمینان از مبادله صحیح و دقیق داده ها ایجاد می شود.

در نظر داشته باشید که صحت داده ها TCP را تضمین می کند، اما عدم ذخیره کردن محدوده های پیغام (ابتدا و انتهای پیغام)در این نوع ارتباط باعث بروز مشکلاتی می شود که در ادامه به بررسی برخی از آنها و راه حل های مقابله خواهیم پرداخت.

پس از پذیرفته شدن داده (Data1)برای ارسال، TCP مدتی آن را در بافر خود نگه داشته و حتی در صورت ارسال داده دیگری توسط برنامه شما، مجدداً در کنار اولین داده نگهداری می گردد. همان طور که پیشتر گفته شد، استریم ها بلوکی از داده ها را ارسال کرده و به همین جهت داده از بافر، بصورت یک بسته ارسال خواهد شد (Data1وData2). در سمت دیگر، ماشین دریافت کننده داده ها تنها یک بسته اطلاعاتی دریافت می کند و چون محدوده های پیغام (ابتدا و انتهای پیغام) در TCP ذخیره نمی شود، ماشین دریافت کننده، بسته های دریافتی را تنها بصورت یک پیغام خواهد دید (Data1+Data2).

یافتن اطلاعات آدرس IP

برای یافتن آدرسIP، چهار روش زیر وجود دارد که در این بخش تنها به روش برنامه نویسی از طریق DNSاشاره خواهیم کرد.

*دستور IPConfig

*استفاده از رجیستری ویندوز

*استفاده از پایگاه داده WMI

*استفاده از DNS (Domain Name System)

سوکت

سوکت به عنوان یک رابط و واسط میان برنامه و پروتوکل شبکه ای مانند TCP، مهمترین بخش از یک برنامه تحت شبکه محسوب می شود. برای درک بهتر عملکرد سوکت، به بررسی مثالی می پردازیم. فرض کنید که یک برنامه نویس، برنامه ای را نوشته که مقادیری را در فایلی درون هارد دیسک ذخیره می کند. واقعیت امر آن است که برنامه نویس تنها از طریق متدهای موجود، اقدام به این عمل کرده و هیچ اطلاعاتی از عملکرد فیزیکی پشت صحنه که سیستم عامل ندارد. بطور مشابه در مورد سوکت نیز، برنامه نویسان بدون توجه به جزئیات مربوط به کارت های شبکه نصب شده بر روی سیستم، نحوه ایجاد بسته های اطلاعاتی و موارد مشابه دیگر، با ایجاد سوکت و فراخوانی متدهایی از آن عملیات مربوط به برقراری اتصال، ارسال، دریافت بسته های اطلاعاتی و پایان دادن به ارتباط را انجام داده و اعمال سطح پایین تر، توسط سیستم عامل انجام می پذیرد.

برنامه های سرویس دهنده و سرویس گیرنده

برنامه سرویس دهنده (Server)

برنامه سرویس دهنده برنامه ای است که بر روی سرور شبکه نصب شده و در خواست هایی را دریافت و پس از پردازش، پاسخ مناسبی به سرویس گیرنده ارسال می کند. در حالت کلی می توان اینگونه نتیجه گرفت که برنامه سمت سرور نمی تواند آغاز کننده ارتباط باشد.

برنامه سرویس گیرنده (Client)

برنامه سرویس گیرنده بر روی Client قرار گرفته و درخواست هایی را به ماشین سرور ارسال می کند و سپس منتظر دریافت پاسخ می ماند. لازم به اشاره است که برنامه سرویس گیرنده را می توان ماشین آغاز کننده ارتباط عنوان کرد زیرا شروع کننده درخواست است.

نحوه مدیریت آدرس های IP.

بر برقراری اتصال بین دو ماشین و اختصاص دادن آدرسIPکارت شبکه خاص به یک سوکت، می بایست یک آدرس IP و شماره پورت (در مدت ارتباط از این پورت برای تبادل اطلاعات استفاده خواهد شد) مشخص نمایید.

انتخاب نوع سوکت ارتباطی

اساساً سوکت ها را به دو بخش اتصال گرا (Oriented-Connection) و بدون اتصال (Connection-Less)می توان تقسیم بندی کرد.

در سوکت های اتصال گرا مانند TCP، برای مبادله داده ها بین دو ماشین، حتماً باید پیشتر، اتصالی برقرار شده باشد. اما در سوکت های بدون اتصال مانندUDP، نیازی به برقراری اتصال نبوده و در عوض، به ازای هر بار ارسال داده به ماشین دیگر، آدرسIP ماشین مقصد می بایست مشخص شود. از آنجایی که سوکت های اتصال گرا از محبوبیت بیشتری در ایجاد برنامه های تحت شبکه برخوردار هستند، لذا در این بخش فقط به بررسی این نوع سوکت ها پرداخته ایم.

سوکت اتصال گرا (Oriented-Connection)

در این بخش، مراحل مختلف ایجاد برنامه های سرویس دهنده و سرویس گیرنده ای که از TPCبرای ارتباط استفاده می کنند، را بررسی کرده و همچنین متدهای مختلف کلاسsocket که برای برقراری ارتباط و همچنین مبادله داده ها وجود دارند، شرح خواهیم داد. برای شروع، ابتدا برنامه سرویس دهنده ساده ای ایجاد خواهیم کرد، سپس به بررسی برنامه سرویس گیرنده می پردازیم.

عملیات برنامه سرویس دهنده (سرور)

مراحل ایجاد برنامه سرویس دهنده به عبارتند از:

*ایجاد سوکت

*مقید کردن(تخصیص)سوکت به یک کارت شبکه (آدرسIP معین)

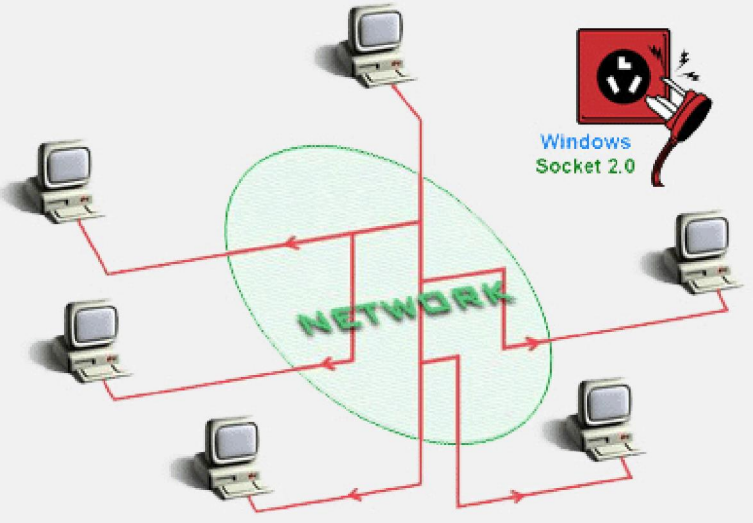
*گوش دادن به درخواست ها

*پذیرش درخواست ها برای برقراری اتصال

*مبادله داده ها

*پایان دا دن به ارتباط و بستن سوکت

کنترل WinSock



کنترل WinSock نسبت به تمام کنترلهای اینترنت در سطح پایینتری قرار دارد . این کنترل امکان ایجاد سرویسهای شبکه ای مبتنی بر پروتکلهای TCP و UDP را مهیا می کند . بعبارت دیگر توسط این کنترل می توان برنامه های کاربردی Server/Client (سرویس گیرنده / سرویس دهنده) ایجاد و با استفاده از پروتکل TCP و یا UDP بین آنها ارتباط برقرار نمود .

با تنظیم خصوصیات و فراخوانی متدهای این کنترل می توانید به راحتی به یک کامپیوتر راه دور متصل شوید و داده ها را در هر دو جهت جابجا نمائید . نمونه کاربرهایی که می توان با این کنترل ایجاد نمود :

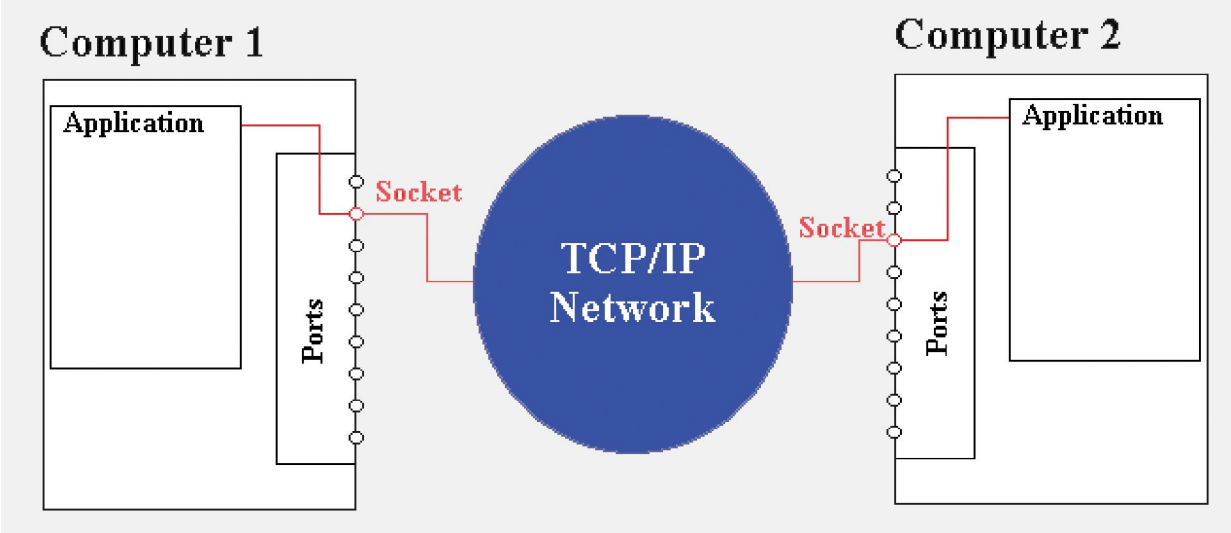
server chat-Client ، Mail client ، Mail server ، Proxy Server ، Network Game ، Port Scanner ، پیاده سازی الگوریتم های موازی و ...

پروتکل کنترل اینترنت (Transfer Control Protocol) اجازه می دهد یک اتصال (Connection) را از طریق سوکت (socket) به یک کامپیوتر راه دور (Remote Computer) ساخته و استفاده کنید . با استفاده از این اتصال ، هر دو کامپیوتر می توانند داده ها را بین خودشان انتقال دهند . برقراری ارتباط از طریق TCP همانند صحبت کردن با تلفن است که باید حتماً اتصالی بین دو کامپیوتر صورت گیرد تا بتوانند با هم ارتباط برقرار کنند . اگر یک برنامه Client می سازید بایستی بدانید که نام یا آدرس IP کامپیوتر Server چیست (Remote Host IP) و همچنین از طریق چه پورتی می توانید به آن متصل شوید (RemotePort) . حال بایستی به آن پورت Connect کنید .

همچنین اگر یک برنامه Server می سازید بایستی پورتی را که روی آن به درخواستها گوش مـی دهیـد مشـخص کنید (LocalPort) و سپس به پورت گوش دهید (Listen) .

زمانیکه یک کامپیوتر Client تقاضای یک اتصال را می دهد Server این درخواست را Accept می کند .

زمانیکه یک اتصال ساخته می شود ، هر دو کامپیوتر می توانند داده را فرستاده و دریافت کنند .

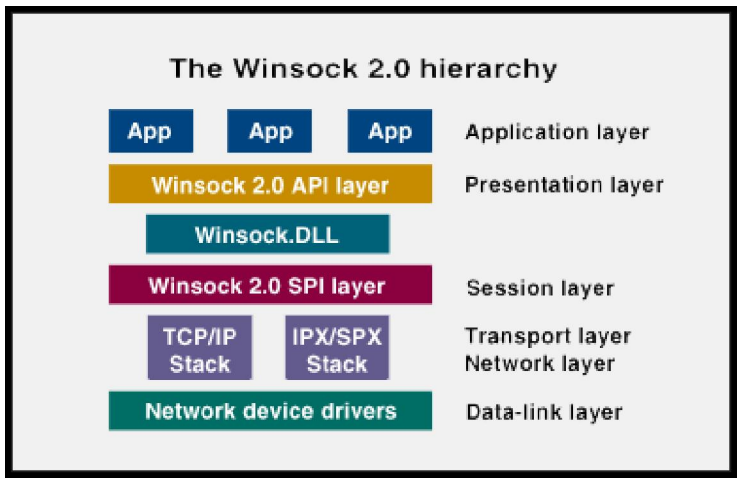


UDP :

پروتکل دیتاگرام کاربر (User Datagram Protocol) پروتکلـی بـدون اتصـال (Connectionless) است . برخلاف TCP ، کامپیوترها نیاز به برپا کردن یک اتصال ندارند بنابراین یک برنامه می تواند یـک client و یـا یک server باشد . برقراری ارتباط در UDP شبیه ارسال نامه از طریق پست است .

برای انتقال داده توسط UDP ابتدا باید Local Port کـامپیوتر Client تنظـیم گـردد . کـامپیوتر Server تنهـا بایستی RemoteHost را برابر آدرس کامپیوتر Client قرار دهد و همچنـین Remote Port را همـان Local Port کامپیوتر Client قرار دهد . سپس دو کامپیوتر می توانند داده ها را بین خود جابجا کنند .

استفاده از کنترل WinSock :



1. انتخابپروتکل: درزماناستفادهازکنترلWinSock اولین کاری کـه بایـد انجـام دهیـد انتخـاب یکـی از پروتکلهـای TCP یا UDP است . طبیعت برنامه ای که شما می سازید نوع پروتکلی را که باید استفاده کنید مشخص می کند . چند سوال زیر به شما کمک می کند که پروتکل مورد نیازتان را انتخاب کنید :

– آیا برنامه شما در زمانیکه داده فرستاده می شود یا دریافت می شود نیاز به اطلاعاتی از طرف Server یا Client دارد ؟ اگر چنین است بایستی یک اتصال TCP قبل از ارسال یا دریافت داده ایجاد شود .

– آیا داده بسیار بزرگ است (مثل تصویر یا فایلهای صوتی) ؟ زمانیکه یک اتصال TCP ساخته می شـود پروتکـل TCP اتصال را باقی نگه می دارد و درستی ارسال داده تضمین شده است . این اتصال در هـر حـال بـه منـابع محاسـباتی بیشتری نیاز دارد و بنابراین پرهزینه تر است .

2. آیا داده متناوب ارسال می شود یا در یک نشست (Session) ارسال خواهد شد ؟ برای مثال اگر شما یک برنامه می سازید که کامپترهای مشخصی را در یک زمان خاص از انجام شـدن عملیـاتی مطلع مـی کنـد پروتکـل UDP مناسـب تـر اسـت . پروتکـل UDP همچنـین بـرای ارسـال مقـادیر کوچـک داده ای مناسـت تـر مـی باشـد . تنظیمپروتکـــل : برایتنظیمپروتکلیکهمیخواهیددربرنامهتاناز آناستفادهکنیددرزمانطراحیبرنامهخاصـــیتProtocol کنترل WinSock را برابر sckTCPProtocol و یا sckUDPProtocol قرار دهیـد . همچنـین مـی توانیـد پروتکل خود را توسط کد زیر تنظیم کنید :

```
WinSock.Protocol=sckTCPProtocol
```

3. مشخص کردن نام کامپیوتان : برای اتصال به کامپیوتر راه دور بایستی آدرس IP و یا نام کامپوتر را بدانید . نام کامپیوتر در Identification/Network/Control Panel موجود است . در صورتیکه می خواهید دو برنامه Client و Server خود را روی یک کامپیوتر تست کنید از آدرس 127 .0.0.1 IP برای هر دو استفاده کنید اما اگر دو برنامه را روی دو کامپیوتر مجزا در شبکه قرار داده اید با اجرای دستور ipconfig در DOS Prompt می توانید آدرس IP کامپیوتر ها را بدست آورید .
4. ایجاداتصال TCP : در زمان ساخت برنامه ای که از پروتکل TCP استفاده می کند ابتدا باید تصمیم بگیرید که این برنامه Client است یا Server . برای ساخت یک برنامه Server بایستی روی یک پورت خاص Listen کنید . زمانیکه Client تقاضای یک اتصال را می دهد ، برنامه Server می تواند آنرا Accept کند و بنابراین اتصال کامل شده است . حال Client و Server می توانند با هم ارتباط داشته باشند .

مراحل زیر ساخت یک سرور چت ساده بر مبنای TCP را نشان می دهد :

- از منوی Project گزینه Components را انتخاب کنید و در لیست Component ها مورد Microsoft WinSock 6.0 را انتخاب کنید .

- یک کنترل WinSock در فرم خود قرار دهید و نام آنرا tcpserver بگذارید

- دو textbox با نامهای txtSendData و txtReceiveData و نیز یکدکمه در فرم قرار دهید .

- کد زیر را در رویداد Form_Load بنویسید :

```
Tcpserver.LocalPort=1000
tcpserver.Listen
```

- زمانیکه درخواستی از طرف Client می آید رویداد ConnectionRequest اجرا می شود . در این رویداد ابتدا باید چک کنید که حالت کنترل بسته باشد . اگر چنین نیست اتصال را قبل از پذیرفتن اتصال جدید ببندید . سپس تقاضا را بر اساس پارامتر requestID می پذیریم :

```
Private Sub tcpserver_ConnectionRequest(ByVal requestID As Long)

If tcpserver.State<>sckClosed Thentcpserver.Close

tcpserver.Accept requestID

End Sub
```

- حال اتصال بین Client و Server برقرار شده است . کد زیر را برای event مربوط به کلیک دکمه Send بنویسید :

```
Tcpserver.SendData txtSendData.text
```

- اگر داده ای از طرف Client بیاید رویداد DataArrival اجرا می شود . کد زیر را برای این رویداد بنویسید :

```
Private Sub tcpserver_DataArrival(ByVal bytesTotalAs Long)
Dim strData As String
tcpserver.GetDatastrData
txtReceiveData.Text =strData
End Sub
```

- کد زیر را برای رویداد Form_Unload بنویسید :

```
Close.Tcpserver
```

مراحل ساخت یک TCP Client بصورت زیر است :

- یک کنترل WinSock در فرم قرار دهید و نام آنرا tcpclient بگذارید .

- دو textbox با نامهای txtsend و txtreceive و نیز یک دکمه با نام sendدر فرم قرار دهید .

- یک دکمه با نام connect در فرم قرار دهید .

- کد زیر را برای متد Form_Load بنویسید :

```
tcpclient.RemoteHost="yourservername"x
tcpclient.RemotePort=1000
```

- کد زیر را برای رویداد کلیک شدن دکمه connect بنویسید :

```
tcpclient.Connect
```

- کد زیر را برای رویداد کلیک شدن دکمه send بنویسید :

```
tctclient.SendData txtsend.Text
```

- کد زیر را برای رویداد DataArrival بنویسید :

```
Private Sub tcpclient_DataArrival(ByVal bytesTotalAs Long)
Dim strData As String
tcpclient.GetDatastrData
txtreceive.Text = strData
End Sub
```

- کد زیر را باری رویداد Form_Unload بنویسید :

```
Tcpclient.Close
```

کدهای فوق یک سیستم Server-Client ساده را نشان می دهد . فایل exe هر دو برنامه را بسازید و آنها را اجرا کنید تا بتوانید سیستم خود را تست کنید .

پذیرفتن بیش ازیک تقاضای اتصال : Server ای که در بالا ساخته شد تنها می تواند تقاضای یک اتصال را بپذیرد . با استفاده از ایجاد یک آرایه از کنترل WinSock می توان چندین تقاضای اتصال را پذیرفت . برای اینکار کافی است یک کپی (instance) از کنترل بسازیم (با تنظیم خاصیت Index) و متد Accept را برای instance جدید بکار ببریم . فرض کنید یک کنترل WinSock با نام sckServer در فرم داریم که خاصیت Index آنرا صفر قرار داده ایم . همچنین یک متغیر intMax از نوع Long تعریف می کنیم که تعداد اتصالات همزمان به Server را نگه می دارد . در event مربوط به Form_Load کد زیر را بنویسید :

```
intMax=0
sckServer(0).LocalPort=1000
sckServer(0).Listen
```

هر بار که تقاضای یک اتصال می رسد کد ابتدا تست می کند که مقدار Index چقدر است . اگر مقدار Index صفر باشد متغیر intMax یکی افزایش می یابد و از intMax برای ساخت یک instance جدید از کنترل استفاده می

شود . حال از این instance برای پذیرفتن تقاضای اتصال استفاده می گردد . برای اینکار کد زیر را برای رویداد ConnectionRequest بنویسید :

```
Private Sub sckServer_ConnectionRequest(Index AsInteger, ByVal requestID
As Long(
If Index =0 Then
intmax =intmax1 +
Load sckServer(intmax(x
sckServer(intmax).(LocalPort0 =
sckServer(Index).(Accept requestID
End If
End Sub
```

ایجاداتصالUDP : ساخت یک برنامه UDP ساده تر از برنامه های TCP است زیرا پروتکل UDP به اتصال نیاز ندارد . در برنامه TCP بالا یک کنترل WinSock بایستی حتماً Listen می کرد و یک کنترل دیگر یک اتصال را توسط متد Connect ایجاد نمود . در عوض پروتکل UDP نیازی به اتصال ندارد . برای ارسال داده بین دو کنترل WinSock سه مرحله بایستی انجام شود :

- پارامتر RemoteHost برابر نام کامپیوتر مقابل است .

- پارامتر RemotePort برابر پارامتر LocalPort کامپیوتر مقابل

- استفاده از متد Bind برای مشخص کردن LocalPort

چون هر دو کامپیوتر از نظر ارتباط مساوی هستند ، این نوع برنامه ها را Peer-to-Peer گویند . برای نمونه از کد زیر برای ساخت یک برنامه chat استفاده می کنیم :

- یک کنترل WinSock در فرم قرار دهید و نام آنرا udppeerA بگذارید .

- خاصیت Protocol آنرا UDPProtocol قرار دهید .

- دو textbox با نامهای txtsend و txtreceive و نیز یک دکمه در فرم قرار دهید .

- کد زیر را برای متد Form_Load بنویسید :

```
udppeerA.RemoteHost=”nameofpeerB”x
udppeerA.RemotePort=1001
udppeerA.Bind1002
```

- کد زیر را برای event مربوط به کلیک دکمه بنویسید :

```
text.SendData txtsend.udppeerA
```

- کد زیر را برای رویداد DataArrival بنویسید :

```
Dim strData as String
udppeerA.GetDatastrData
txtreceive.Text=strData
```

برای ساخت UDP peerB مشابه مراحل بالا عمل کنید فقط خاصیت RemoteHost آنرا نام کامپیوتر PeerA و خاصیت RemotePort آنرا 1002 و خاصیت Bind آنرا 1001 قرار دهید .

مشکلات TCP

در برنامه های تحت شبکه ای که از پروتکل TCP در ارتباط خود با ماشین دیگر استفاده می کنند، برنامه نویسان با دو مشکل اساسی زیر مواجه هستند:

*استفاده نامناسب و دستکاری ناصحیح بافر داده

*اندازه نامناسب پیغام ها در شبکه

دقت نظر داشته باشید که در برنامه هایی که ذکر گردید، همه پیغام ها قالب ثابت و مشخصی داشته و اندازه آنها کنترل شده بود، اما در اصل هنگام برقراری ارتباط سرویس دهنده با سرویس گیرنده، هیچ یکی از آنها شناختی در مورد طول یا نوع داده های ارسالی صرف مقابل ندارند. حال این سوال مطرح می شود که هنگام دریافت اطلاعاتی که حجمشان بیش از حجم تعیین شده است، چه باید کرد؟

خب استفاده از بافر داده بزرگ یا بافر داده کوچک هر یک مزایا و معایب خود را دارند که بررسی آنها خارج از حوصله این بخش است ولی به طور خلاصه بدانید که بر حسب عملکرد برنامه، می بایست سایز بافر داده را تعیین کنیم. ضمناً مشکل دیگر ارتباط TCP، عدم نگهداری محدوده های پیغام در این نوع ارتباط است که برای بر طرف کردن آن سه روش وجود دارد:

*ارسال پیغام با طول ثابت

*ارسال اندازه پیغام به همراه پیغام

*استفاده از سیستم نشانه گذاری برای جدا کردن پیغام ها

حال از آنجایی که روش سوم از سه روش یاد شده، کاراتر بوده و ضریب اطمینان بالاتری دارد، در ادامه بـه بررسـی آن خواهیم پرداخت.

برنامه های سرویس دهنده و سرویس گیرنده

برنامه سرویس دهنده(Server)

برنامه سرویس دهنده برنامه ای استکه بر روی سـرور شـبکه نصـب شـده و در خواسـت هـایی را دریافـت و پـس از پردازش، پاسخمناسبی به سرویس گیرنده ارسال می کند. در حالت کلی می توان اینگونه نتیجه گرفت کهبرنامـه سـمت سرور نمی تواند آغاز کننده ارتباط باشد .

برنامه سرویس گیرنده(Client)

برنامه سرویس گیرنده بر روی Client قرار گرفته و درخواست هایی را به ماشین سـرور ارسـال مـی کنـد و سـپس منتظردریافت پاسخ می ماند. لازم به اشاره است که برنامه سرویس گیرنده را می توان ماشینآغاز کننده ارتباط عنوان کرد زیرا شروع کننده درخواست است.

عملیات برنامه سرویس دهنده (سرور)

مراحل ایجاد برنامه سرویسدهنده عبارتند از:

*ایجاد سوکت

*مقید کردن(تخصیص)سوکت به یک کارت شبکه(آدرس IPمعین)

*گوش دادن به درخواست ها

*پذیرش درخواست ها برای برقراریاتصال

*مبادله داده ها

*پایان دا دن به ارتباط و بستن سوکت

عملیات برنامه سرویس گیرنده

مراحل ایجاد برنامه سرویسگیرنده عبارتند از:

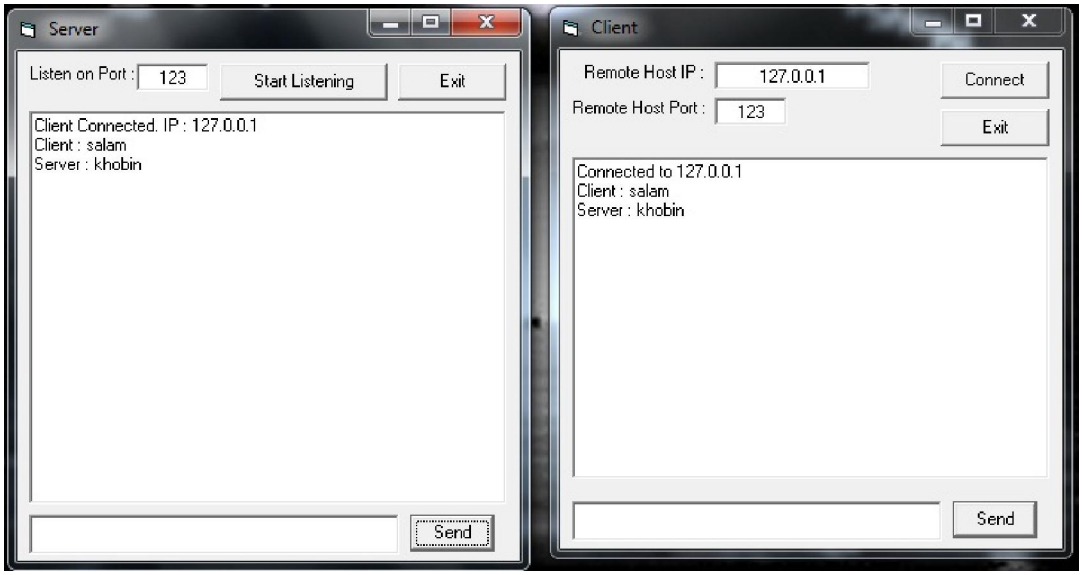
*ایجاد سوکت

*متصل شدن بهسرورها

*ارسال و دریافت داده ها

*پایان دادن به ارتباط و بستن سوکت

در زیر به یک نمونه برنامه با زبان VB با استفاده از سوکت اتصال گرا برای سمت سرویس گیرنده و سرویس دهنده اشاره می کنیم.



1. کنترلهای مربوط به سمت سرویس دهنده:

a. کنترل winsocket (sock1)

b. سه عدد کنترل Text Box(txtsend,txtrecive,txtport)

c. سه عدد کنترل Button(btnexir,btnsend,btnlisten)

کد مربوط به سرویس دهنده :

```
Private Sub bntListen_Click()
On Error GoTo t
sock1.Close
sock1.LocalPort = txtPort
sock1.Listen
Exit Sub
t:
MsgBox Error :& Err.Description, vbCritical
End Sub
Private Sub bntExit_Click()
End
End Sub
Private Sub bntSend_Click()
On Error GoTo t
sock1.SendData txtSend
txtLog = txtLog & Server :  & txtSend & vbCrLf
txtSend =
Exit Sub
t:
MsgBox Error :& Err.Description
sock1_Close
```

```
End Sub

Private Sub sock1_Close()

sock1.Close

txtLog = txtLog & *** Disconnected & vbCrLf

End Sub

Private Sub sock1_ConnectionRequest(ByVal requestID As Long)

If sock1.State <> sckClosed Then sock1.Close

sock1.Accept requestID

txtLog = Client Connected. IP :& sock1.RemoteHostIP & vbCrLf

End Sub

Private Sub sock1_DataArrival(ByVal bytesTotal As Long)

Dim dat As String

sock1.GetData dat, vbString

txtLog = txtLog & Client :  & dat & vbCrLf

End Sub

Private Sub sock1_Error(ByVal Number As Integer, Description As String, ByVal
Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal
HelpContext As Long, CancelDisplay As Boolean)

txtLog = txtLog & *** Error :  & Description & vbCrLf

sock1_Close

End Sub
```

2- کنترلهای مربوط به سمت مشتری

a. منترل winsocket(sock1)

b. چهار عدد کنترل TextBox(txtsend,txtrecive,txtport,txtip)

c. سه عدد دگمه کنترل (bntsend,bntexit,bntconnect)

کد سمت مشتری:

```
Private Sub bntConnect_Click()
On Error GoTo t
sock1.RemoteHost = txtIP
sock1.RemotePort = txtPort
sock1.Connect
Exit Sub
t:
MsgBox Error :& Err.Description, vbCritical
End Sub
Private Sub bntExit_Click()
End
End Sub


Private Sub bntSend_Click()
On Error GoTo t
sock1.SendData txtSend
txtLog = txtLog & Client :  & txtSend & vbCrLf
txtSend =
Exit Sub
t:
MsgBox Error :& Err.Description
sock1_Close
End Sub
```

```
Private Sub sock1_Close()

sock1.Close

txtLog = txtLog & *** Disconnected & vbCrLf

End Sub


Private Sub sock1_Connect()

txtLog = Connected to  & sock1.RemoteHostIP & vbCrLf

End Sub


Private Sub sock1_DataArrival(ByVal bytesTotal As Long)

Dim dat As String

sock1.GetData dat, vbString

txtLog = txtLog & Server :  & dat & vbCrLf

End Sub

Private Sub sock1_Error(ByVal Number As Integer, Description As String, ByVal
Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal
HelpContext As Long, CancelDisplay As Boolean)

txtLog = txtLog & *** Error :  & Description & vbCrLf

sock1_Close

End Sub
```

تمرین اول برنامه نویسی – برنامه browser ساده بزبان vb

برنامه Internet Explorer یا exe.iexplore در واقع برنامه کوچکی است که وظیفه اصلی آن ایجاد چارچوبی برای بهم پیوستن عناصر مختلف است و این عناصر هستند که وظایف اصلی مثل load کردن صفحات وب ،

اجرای کدهای Html و غیره را انجام می دهند . اصلی ترین عنصری که مستقیماً توسط exe.iexplore استفاده می شود کنترل Webbrowser (موجود در فایل dll.shdocrw) می باشد . وظیفه این فایل dll ، عبارت است از حرکت بین صفحات وب ، مدیریت تاریخچه صفحات دیده شده و غیره . این فایل خود از فایل دیگری بنام dll.Mshtml استفاده می کند که وظیفه آن بررسی و اجرای فایلهای html است . مایکروسافت به برنامه نویسان این امکان را داده که بتوانند در برنامه هایشان از کنترل webbrowser استفاده کنند . با استفاده از این کنترل می توان به سادگی یک مرورگر وب تقریباً کامل ساخت .

خصوصیات کنترل Webbrowser :

Webbrowser علاوه بر خواص استانداردی مثل width، height و ...خواص زیررادارد :

–Busy : اگر در حال load کردن یک صفحه یا در حال جستجو در وب باشد این خاصیت True است . توسط متد Stop می توان عملیات جاری را متوقف کرد .

–Container : ارجاع به شی نگهدارنده کنترل webbrowser

–Document : ارجاع به صفحه html فعلی . برای کار با این صفحه html می توان از خواص و متدهابی شیDocument استفاده کرد .

–LocationName : حاوی آدرس محلی است که اکنون در کنترل webbrowser، load شده است . اگر این محل یک صفحه html باشد عنوان آن صفحه خواهد بود و اگر این محل یک فایل در شبکه باشد مسیر کامل آن فایل خواهد بود .

–LocationURL : حاوی url محلی است که فعلاً در کنترل webbrowser، load شده است .

–Offline : اگر کنترل webbrowser در حالت عدم اتصال باشد مقدار آن True و در غیراینصورت False است .

–Parent : فرمی را نشان می دهد که کنترل webbrowser در آن قرار دارد .

–ReadyState : وضعیت کنترل webbrowser را برمی گرداند .

متدهای کنترل webbrowser : این متدها مربوط به مرور در صفحات وب هستند :

GoBack : در لیست تاریخچه url ها ، یکی به عقب برمی گردد .

GoForward : در لیست تاریخچه url ها ، یکی به جلو می رود .

GoHome : به homepage مرورگر می رود .

Navigate : به یک url یا فایل می رود . ساختار این متد بصورت زیر است :

Navigate URL[Flags,][TargetFrameName,][PostData,][Headers]x

URL آدرس مقصد می باشد .Flags نحوه باز شدن آدرس مقصد را تعیین می کند . اگر این پارامتر ذکر نشود آدرس جدید در پنجره فعلی باز خواهد شد و به لیست تاریخچه اضافه شده و اگر کپی آن در temperorycache موجود باشد از آنجا خوانده می شود . مقادیر پارامتر Flags عبارتند از :

- NavOpenInNewWindow : آدرس جدید را در پنجره جدیدی باز می کند .
- NavNoHistory : به لیست تاریخچه اضافه نمی شود بلکه جایگزین صفحه فعلی می شود .
- NavNoReadFromCache : صفحه جدید از cache خوانده نمی شود .
- NavNoWriteToCache : صفحه جدید روی cache نوشته نمی شود

Event های کنترل webbrowser : این event ها مربوط به مرور در وب و تغییر حالت آن هستند :

CommandStateChange : برای فعال یا غیرفعال کردن دکمه های Forward و Back در مرورگر استفاده می شود . شکل کلی فراخوانی این event بصورت زیر است :

Private Sub WebBrowser1_CommandStateChange(ByVal Command As Long, ByVal Enable As Boolean)

که command فرمانی است که حالت فعال آن تغییر کرده است و دو مقدار می گیرد : 1 و 3 که بترتیب معادل فرمانهای GoForward و GoBack هستند .

Enable فعال یا غیرفعال بودن فرمان را تعیین می کند .

DocumentComplete : این event زمانی فعال می شود که صفحه در حال load شدن به حالت ReadyState_Complete برود . شکل کلی فراخوانی این event بصورت زیر است :

Private Sub WebBrowser1_DocumentComplete(ByVal pDisp As Object, URL AsVariant)

که pDisp ارجاعی به کنترل webbrowser است که event در آن رخ داده است و URL آدرس صفحه در حال load شدن است

DownloadBegin : این event در آغاز حرکت به صفحه جدید روی می دهد و هیچ پارامتری نمی گیرد . مرورگر می تواند در این eventپیغامی برای شروع عملیات جدید نشان می دهد .

DownloadComplete : این event در پایان عملیات یا در صورت انصراف کاربر یا بروز خطا روی می دهد .

ProgressChange : با بروز هر تغییری در وضعیت load ، این event روی می دهد . شکل کلی فراخوانی آن بصورت زیر است :

Private Sub WebBrowser1_ProgressChange(ByVal Progress As Long, ByVal ProgressMax As Long)

که Progress نشان دهنده پیشرفت عملیات (بایتهای load شده) است . پارامتر ProgressMax تعداد کل بایتهایی که باید load شوند را نشان می دهد بنابر این :

load=درصد پیشرفت عملیات (Progress/ProgressMax*100)

مثالی دیگر از Browser

یک مثال ساده از ایجاد یک مرورگر اینترنت (Internet Explorer) را در محیط ویژوال بیسیک نشان داده می شود. البته در اینجا فقط شما رو با مبانی کار آشنا میکنم (جسارت نباشه منظورم کسانیه که مثل شما حرفه ایی نیستن) و بعد خودتون میتونین پوسته های دلخواهتون رو برای مرورگر بسازین.

ساختن یک کاوشگر اینترنت خیلی آسونه. و فقط شامل چند خط کد سادَس.خوب ابتدا وارد محیط ویژوال بیسیک بشین و یک پروژه جدید از نوع استاندارد بسازین.

ابتدا باید محیط گرافیکی (GUI) رو آماده کنیم. برای اینکار داخل فرم اشیا زیر رو اضافه کنید:

1- پنج تا دگمه معمولی Command Button

2- یک دونه کاوشگر اینترنت Web Browser

3- یک عدد جعبه متن معمولی Text Box

نکته: برای اضافه کردن WebBrowser روی جعبه ابزار راست کلیک کنید و گزینه Add Component رو انتخاب کنید توی صفحه ای که باز میشه کنار تیک کنار اسمش رو بزنین.

حالا باید کدهای مربوط به دگمه ها رو بنویسیم. توجه کنید که اون جعبه متنی رو که به فرم اضافه کردین محل وارد کردن آدرس (URL) سایت مورد نظر هستش. اول باید وظیفه هرکدوم از کدها رو مشخص کنیم تا کدهای مربوط به اونها رو بنویسیم. همونطور که در بالا گفتم باید پنج تا دگمه روی فرم داشته باشیم که بترتیب نام و وظیفه هرکدوم رو میگم:

1- GO برای رفتن به آدرس تایپ شده در کادر متنی

2- Back برای برگشت به صفحه قبل

3- Forward برای رفتن به صفحه بعد

4- Refresh برای لود کردن مجدد صفحه

5- Stop برای توقف عمل لود کردن صفحه

حالا برای صرفه جویی در وقت موقع تایپ کردن کدها ایم Web Browser رو به WB تغییر بدین. در زیر کدهای مربوط به رویداد لود فرم اصلی رو می بینین:

```
Private Sub Form_Load()
wb.Navigate \"http://www.1KOLBE.blogfa.com\"
End Sub
```

کد بالا باعث میشه که تا برنامه رو ایجاد میکنید مرورگر اینترنت فرمتون وبلاگ من رو لود کنه، البته می تونین بجای آدرسه وبلاگ من هر آدرس دیگه ایی بنویسین. حتی اگه حرفه ای باشین می تونین توی فرمتون یک کلید دیگه اضافه کنین و اسمشرو بزارین HomePage بعدش کدی رو بنویسین که وقتی این دگمه رو کلیک میکنید آدرس مجود در کادر متنی تون رو یه جایی توی رجیستری ویندوز ذخیره کنه و بعدا که دوباره برنامه رو اجرا میکنید اون آدرس رو لود کنه. خوب حالا کد های مربوط به دگمه ها رو توی برنامه وارد کنید:

دگمه Refresh

```
Private Sub Command5_Click()
wb.Refresh
End Sub
```

دگمهStop

```
Private Sub Command3_Click()
wb.Stop
End Sub
```

دگمهBack

```
Private Sub Command1_Click()
wb.GoBack
End Sub
```

دگمهForward

```
Private Sub Command2_Click()
wb.GoForward
End Sub
```

دگمهGo

```
Private Sub Command6_Click()
wb.Navigate Text1.Text
End Sub
```