



برنامه نویسی سوکت های TCP/IP در C

نویسندگان:

مایکل جی. داناهاو

کِنِثِ اِل. کالورت

مترجم:

غلامرضا صابری تبریزی

کتابراه

سرشناسه	: داناھو مایکل جی. Donahoo, Michael J
عنوان و نام پدیدآور	: برنامه‌نویسی سوکت‌های TCP/IP در C نویسندگان مایکل جی. داناھو، کنث ال. کالورت.
مشخصات نشر	: مترجم غلامرضا صابری تبریزی.
مشخصات ظاهری	: تهران انتشارات دانشگاهی کیان (کیان رایانه سبز) ۱۳۹۱.
شابک	: ۲۴۰ص: مصور. ۹۷۸-۶۰۰-۶۰۲۱-۹۱-۱
وضعیت فهرست‌نویسی	: فیا
یادداشت	: عنوان اصلی: TCP/IP sockets in C : practical guide for programmers.
یادداشت	: کتابنامه: ص. ۲۳۵.
موضوع	: رابط‌های برنامه‌سازی کاربری.
موضوع	: تی. سی. پی. آی. پی (پروتکل شبکه کامپیوتری)
موضوع	: سی (زبان برنامه‌نویسی کامپیوتر)
شناسه افزوده	: کالورت، کنث ال. Calvert, Kenneth L.
شناسه افزوده	: صابری تبریزی، غلامرضا، ۱۳۶۷-، مترجم.
رده‌بندی کنگره	: QAV777/22 ۱۳۹۱
رده‌بندی دیویدی	: ۰۰۵/۷۱۳۷۶۲
شماره کتابشناسی ملی	: ۲۸۹۹۳۳۲



انتشارات دانشگاهی کیان

نام کتاب	: برنامه‌نویسی سوکت‌های TCP/IP در C
ناشر	: دانشگاهی کیان (کیان رایانه سبز)
مؤلفان	: مایکل جی. داناھو- کنث ال. کالورت.
مترجم	: غلامرضا صابری تبریزی
چاپ اول	: ۱۳۹۱
تیراژ	: ۱۰۰۰ جلد
چاپ و صحافی	: گنج شایگان
قیمت	: ۹۰۰۰ تومان (همراه با CD)
شابک	: ۹۷۸-۶۰۰-۶۰۲۱-۹۱-۱
ISBN	: 978-600-6021-91-1

کلیه حقوق برای ناشر محفوظ است.
تکثیر تمامی یا قسمتی از این اثر به صورت حرفه‌جینی یا چاپ مجدد، چاپ افست، فتوکپی و انواع دیگر چاپ ممنوع است و پیگرد قانونی دارد.

مرکز پخش:

انتشارات دانشگاهی کیان: خ انقلاب - ۱۲ فروردین - کوچه نوروژ - پلاک ۲۷ - طبقه اول

تلفن: ۶۶۴۱۶۴۴۶-۶۶۴۰۶۸۳۴-۶۶۴۱۱۷۱۵

خرید Online از طریق سایت www.kianpub.com

SMS : ۳۰۰۰۲۲۱۴۴۱

سخنی با خوانندگان

«سپس، به کاتبان و نویسندگان بنگر و بهترین آن‌ها را بر کارهای خود بگمار...
کاتبان و نویسندگانی برگزین که قدر خود را بشناسند، چون کسی که به قدر خود شناخت
ندارد، دیگران را هم نمی‌شناسد.»
«برگرفته از نامه‌ی ۵۳ نهج البلاغه به مالک‌اشتر»

اگرچه نوشتن و پرداختن زکات علم از توصیه‌های اکید بزرگان و گواه بر کرامت اهل دانش است، اما امروزه پرداختن به انگیزه‌ها و اهداف نوشتن بیشتر جلوه می‌کند. بی‌شک این‌که چه کسی می‌نویسد مهم نیست، اما این‌که چرا و به چه پشتوانه‌ای می‌نویسد، درخور تأمل است. ما معتقدیم که چاپ روزافزون کتاب‌های به اصطلاح «زرد» که خالی از هرگونه نوآوری و بی‌توجه به استانداردهای چاپ کتاب و نیازهای مخاطبان است، حاصل تفکر بازاری مستولی بر جامعه‌ی نشر است. بی‌پرده آن‌که عنوان پر زرق و برق، دستاویز قرار دادن مضمون‌های نو با هدف فروش بالا و طویل کردن سیاهه‌ی سابقه‌ی علمی، نمی‌تواند دلیل محکمی برای چاپ و نشر کتابی باشد که خواننده‌ی مشتاق با صرف هزینه‌های نه چندان کم آن را تهیه می‌کند؛ به امید آن که چیزی از آن بیاموزد.

باید پذیرفت که انگیزه‌ی نوشتن کم از محتوای نوشته نیست و بین این دو رابطه‌ای مستقیم برقرار است. اگر انگیزه از نوشتن، تولید دانش باشد، بی‌شک نویسنده از قلم بی‌محتوا و کم‌عمق پرهیز می‌کند و اگر دغدغه‌ی دانش و فرهنگ زخم‌خورده در میان باشد، ناشر تنها به عنوان پرطمطراق بسنده نمی‌کند.

و چقدر امروزه، فرهنگ و دانش این مرز بوم که گرفتار آفت بی‌انگیزگی و زخم هوس است، نیازمند ناشران و نویسندگانی است که نیت‌شان کمک به رشد دانش و ارتقای فرهنگ جامعه است و به راستی که التیامی بر این درد نیست مگر نویسندگانی که قدر خود و دیگران را می‌دانند و خوب می‌فهمند که کتاب، ابزار سودجویی‌های مغرضانه نیست و می‌کوشند تا خود را از هرگونه شهوت نام و رسم و ثروت تهی کنند.

انتشارات کیان رایانه‌ی سبز خود را بری از عیب و خطا نمی‌داند، اما همواره بیش از پیش می‌کوشیم تا در راستای تولید علم و نشر کتاب‌های پرمحتوا، دست نویسندگانی که انگیزه‌ی پاک دارند را فشرده و در کنارشان باشیم و از خداوند متعال می‌خواهیم که در این مسیر صعب و پرخطر در سایه‌ی لطف و عنایت خود از آن‌چه به عهده‌ی ما نهاده شده، سربلند و پیروز برآییم.

انتشارات کیان رایانه‌ی سبز

مقدمه مترجم

امروزه با گسترش هرچه بیشتر استفاده از اینترنت، به وجود برنامه‌هایی نیاز داریم که بتوانند با استفاده از شبکه با یکدیگر ارتباط برقرار کنند.

کتاب حاضر مرجعی است برای اینکه علاقه‌مندان بتوانند با استفاده از آن به یادگیری مقدمات برنامه‌نویسی سوکت بپردازند. به این منظور ابتدا در سال ۱۳۸۶ مقاله «Beej's guide to network programming» را ترجمه کردیم و در اختیار مخاطبان قرار دادیم. این مقاله در این سال‌ها با استقبال بسیاری روبه‌رو شد و ما را بر آن داشت تا مرجع جامع‌تری را در این زمینه در اختیار مخاطبان قرار دهیم. نتیجه این تلاش کتاب حاضر است که در جهت فراهم آوردن مرجعی جامع‌تر و به‌روزتر برای مخاطبان تهیه شده است. این کتاب ترجمه ویرایش دوم «TCP/IP Sockets in C: Practical Guide for Programmers» است که انتشارات Morgan Kaufmann در سال ۲۰۰۹ به چاپ رسانده است. در ترجمه کتاب سعی کرده‌ایم سودمندترین مطالب را ارائه دهیم بنابراین از ترجمه فصل ۸ کتاب، که به بررسی کتابخانه‌ای خاص برای برنامه‌نویسی سوکت در C++ می‌پردازد، صرفنظر کردیم.

مطالب این کتاب برای علاقه‌مندان به یادگیری چگونگی نوشتن برنامه‌های مبتنی بر سوکت در سیستم عامل لینوکس با استفاده از زبان C و دانشجویانی که در حال گذراندن واحدهای برنامه‌نویسی شبکه و برنامه‌نویسی سیستمی هستند بسیار سودمند خواهد بود. گفتنی است بیشتر مفاهیم موجود در این کتاب در سایر سیستم‌های عامل نیز صادق است و تفاوت‌ها بسیار اندک است. همچنین توضیحاتی را که مترجم لازم دیده است، در پاورقی‌ها با ذکر نشان -م. مشخص کرده است. لوح فشرده‌ای نیز همراه کتاب عرضه شده است که حاوی نسخه‌ی پی‌دی‌اف کتاب و کدهای مندرج در آن است.

سپاسگزار خواهیم بود اگر نظرات و انتقادات خود را درباره این کتاب به آدرس پست الکترونیکی مترجم ارسال نمایید. صمیمانه در انتظار نظرات و انتقادات سازنده شما عزیزان هستیم. در انتها جای دارد از این بزرگواران سپاسگزاری کنم: خواهر عزیزم زهرا که زحمت ویرایش کتاب بر عهده ایشان بود و خود بهتر می‌داند که برای من چه کرده است، سرکار خانم طاهره سادات میراحمدی که با دقت و حوصله متن کتاب را حروفچینی کردند؛ همچنین از سرکار خانم پریسا کیانی که با بزرگواری زحمت طراحی جلد را بر عهده گرفتند قدردانی می‌کنم.

غلامرضا صابری تبریزی

reza_sabery_89@yahoo.com

<http://www.gstsoftware.blogfa.com>

مقدمه نویسندگان

مقدمه ویرایش دوم

وقتی ویرایش اول این کتاب به چاپ می‌رسید، برگزاری دوره‌های برنامه‌نویسی شبکه در دانشگاه‌ها رواج چندانی نداشت. حال باور رواج این چینی این دوره‌ها و پذیرفته‌شدن فواید آموزشی آن، در زمانی که نقش اینترنت در زندگی روزمره‌مان اینقدر پررنگ شده، برایمان دشوار است. اگرچه درحال حاضر زبان‌های برنامه‌نویسی دیگری نیز وجود دارند که امکان دستیابی به اینترنت را فراهم می‌آورند، هنوز هم عده کثیری به یادگیری چگونگی استفاده از سوکت‌های برکلی^۱ در زبان C علاقه دارند. API سوکت در دهه هشتاد و در دانشگاه برکلی برای برنامه‌نویسی شبکه در یونیکس BSD ابداع شد- یکی از اولین پروژه‌هایی که در حال حاضر به آن‌ها پروژه‌های اپن‌سورس^۲ (متن باز) می‌گویند.

اینترنت و API سوکت در دنیایی رشد کردند که خانواده پروتکل‌های^۳ مختلفی در آن با هم به رقابت می‌پرداختند- پروتکل‌هایی نظیر IPX، AppleTalk، DECNet، OSI، SNA و TCP/IP- و سوکت‌ها برای پشتیبانی از تمامی آن‌ها ایجاد شده بودند. وقتی ویرایش اول این کتاب نوشته می‌شد پروتکل‌های کمتری به صورت گسترده مورد استفاده بودند و در حال حاضر این تعداد کمتر هم شده است. درحال همان طور که در ویرایش اول پیش‌بینی می‌کردیم، API سوکت برای آن دسته از افرادی که به طراحی و ساخت برنامه‌های توزیع‌شده^۴ می‌پردازند که از اینترنت (TCP/IP) استفاده می‌کند هنوز هم حائز اهمیت است. در حال حاضر این API امکان برنامه‌نویسی برای نسخه جدید پروتکل IP (IPv6) را که تقریباً تمامی سکوها^۵های رایج از آن پشتیبانی می‌کنند نیز فراهم آورده است.

مخاطبان

مخاطبان این کتاب به دو دسته کلی تقسیم می‌شوند: دانشجویانی که در حال گذراندن واحدهای مقدماتی شبکه‌های کامپیوتری^۶ به همراه اصول برنامه‌نویسی مربوط به آن هستند و پژوهشگرانی که به نوشتن برنامه‌هایی تمایل دارند که از طریق اینترنت ارتباط برقرار می‌کنند.

1. Berkley Sockets
2. Open Source
3. Protocol Families
4. Distributed
5. Platform
6. Computer Networks

به یاد داشته باشید که مطالعه این کتاب به تنهایی شما را به یک برنامه‌نویس خیره در زمینه شبکه تبدیل نخواهد کرد. هدف این کتاب فراهم آوردن مقدمه‌ای است تا شما را قادر سازد در این زمینه تجربه و دانش اولیه‌ای کسب کنید تا بدین‌سان بتوانید یادگیری خود را ادامه دهید.

پیش‌نیازها

در این کتاب فرض کرده‌ایم شما با اصول اولیه برنامه‌نویسی با استفاده از زبان C و سیستم‌های مبتنی بر یونیکس^۱ آشنایی دارید. بنا را بر این گذاشته‌ایم که مفهوم اشاره‌گرها^۲ و تبدیل انواع داده‌ای^۳ را می‌شناسید و آشنایی مختصری با نمایش داده‌ها به صورت دودویی^۴ دارید. برخی از مثال‌ها باید در قالب فایل‌هایی به صورت جداگانه کامپایل شوند، فرض کرده‌ایم شما توانایی انجام این را نیز دارید.

به علاوه، باید با مفهوم فرآیند^۵ فضای آدرس^۶ آرگومان‌های خط فرمان^۷ مختوم کردن^۸ برنامه‌ها و ورودی/خروجی با استفاده از فایل در سیستم‌های مبتنی بر یونیکس نیز آشنایی داشته باشید.

سکوی مورد استفاده و قابلیت حمل^۹ کدهای مثال

مثال‌های بررسی شده در این کتاب مبتنی بر یونیکس هستند. در زمان نوشتن کتاب چندین نفر از ما خواستند تا کدهای مبتنی بر ویندوز را نیز در کتاب درج کنیم. متأسفانه این امر به دلایلی نظیر بالا رفتن تعداد صفحات کتاب (و به تبع آن قیمت) امکان‌پذیر نبود.

کسانی که تنها به سکوهایی مبتنی بر ویندوز دسترسی دارند، توجه داشته باشند که مثال‌های موجود در فصل‌های اول کتاب به تغییرات بسیار اندکی نیاز دارند تا بتوان آن‌ها را در ویندوز نیز اجرا کرد. (باید فایل‌های سرآیند را تغییر داده و توابع بخصوصی را در ابتدا و انتهای برنامه‌ها فراخوانی کنید). اکثر مثال‌های دیگر نیز به تغییرات اندکی نیاز دارند. با این حال برخی از مثال‌ها به دلیل وابستگی به مدل برنامه‌نویسی یونیکس در ویندوز قابل استفاده نیستند. نسخه‌های مبتنی بر ویندوز سایر مثال‌ها و دستورالعمل‌های مربوط به چگونگی تغییر کدها برای سازگاری با ویندوز

1. Unix
2. Pointers
3. Type Casting
4. Binary
5. Process
6. Address Space
7. Command-line argument
8. Termination
9. Portability

را می‌توانید از وبسایت کتاب به آدرس <http://www.mkp.com/socket> دریافت کنید. توجه داشته باشید که تمامی مثال‌های موجود را می‌توانید با اندکی تغییرات با استفاده از کتابخانه 'Cygwin' که به صورت آنلاین موجود است در ویندوز نیز اجرا کنید.

در این ویراست از استاندارد C99 استفاده کرده‌ایم. این نسخه از زبان C به‌وسیله اکثر کامپایلرها پشتیبانی می‌شود و امکانات بسیاری را برای افزایش خوانایی کد فراهم آورده است - مانند: توضیحات تک‌خطی، انواع عددی صحیح با طول ثابت، امکان تعریف متغیرها در هر مکانی از بلاک و

ما در کدهایمان از اضافات "Basic Socket Interface Extensions for IPv6" استفاده کرده‌ایم. یکی از این اضافات رابط جدید سیستم نام^۲ است. به دلیل وابستگی کدهای ما به این رابط (getaddrinfo) ممکن است کدهایمان در برخی سکوها قدیمی‌تر اجرا نشوند.

تمامی کدهای موجود در سیستم‌های مبتنی بر یونیکس و مکینتاش^۳ تست شده‌اند (و باید بتوانید بدون هیچ‌گونه تغییری آنها را کامپایل و اجرا کنید). با این حال متأسفانه مکان قرارگیری فایل‌های سرآیند (h) و برخی کتابخانه‌ها در تمامی سیستم‌ها استاندارد نیست و ممکن است برای اجرای مثال‌ها در سکوی مورد استفاده‌تان به اندکی تغییر نیاز داشته باشید. به علاوه برخی از تنظیمات مربوط به سوکت نیز از سکویی به سکوی دیگر متفاوت‌اند؛ بنابراین سعی کرده‌ایم تنظیماتی را بررسی کنیم که در تمامی سکوها وجود دارند. برای اطلاع از تنظیمات مختص به سکوی مورد استفاده‌تان می‌توانید به صفحات راهنمای آن سکو (منظور همان صفحات man است. برای اطلاعات بیشتر درباره این صفحات می‌توانید از دستور `man man` استفاده کنید یا در اینترنت جستجو نمایید) مراجعه کنید.

در پایان باید یادآور شویم که کدهای موجود برای اهداف آموزشی نوشته شده‌اند و برای حفظ سادگی دارای کیفیت و پارامترهای کدهای تجاری نیستند.

مایکل جی. داناو

کنث ال. کالورت

۱. مجموعه‌ای از ابزارهای نرم‌افزاری آزاد است که به نسخه‌های متفاوت از میکروسافت ویندوز این امکان را می‌دهد که مانند سیستم‌عامل یونیکس عمل نمایند.م.

2. Name System

3. MAC OS

فهرست مطالب

فصل اول: سرآغاز

- ۱-۱ شبکه‌ها، بسته‌ها و پروتکل‌ها ۱۶
- ۲-۱ آدرس‌ها ۲۰
- ۱-۲-۱ نوشتن آدرس‌های IP ۲۰
- ۲-۲-۱ استفاده از هر دو نسخه IP ۲۱
- ۳-۲-۱ شماره‌های درگاه ۲۲
- ۴-۲-۱ آدرس‌های خاص ۲۳
- ۳-۱ نام‌ها ۲۴
- ۴-۱ مشتری‌ها و سرویس‌دهنده‌ها ۲۵
- ۵-۱ سوکت چیست؟ ۲۶
- تمرین‌ها ۲۸

فصل دوم: مبانی سوکت‌های TCP

- ۱-۲ مشتری مبتنی بر IPv4 و TCP ۳۰
- ۲-۲ سرویس‌دهنده مبتنی بر TCP و IPv4 ۳۸
- ۳-۲ ایجاد و نابودکردن سوکت‌ها ۴۵
- ۴-۲ مشخص کردن آدرس‌ها ۴۶
- ۱-۴-۲ آدرس‌های عام ۴۷
- ۲-۴-۲ آدرس‌های IPv4 ۴۷
- ۳-۴-۲ آدرس‌های IPv6 ۴۸
- ۴-۴-۲ ذخیره‌سازی آدرس‌ها به صورت عام ۴۹
- ۵-۴-۲ تبدیل آدرس‌ها به قالب‌های دودویی و رشته‌ای ۵۰
- ۶-۴-۲ به دست آوردن آدرس‌های مرتبط با یک سوکت ۵۱
- ۵-۲ متصل کردن سوکت ۵۲
- ۶-۲ بایند شدن به یک آدرس ۵۲
- ۷-۲ رسیدگی به ارتباطات ورودی ۵۴
- ۸-۲ ارسال و دریافت داده‌ها ۵۶

۲-۹ استفاده از IPv6 ۵۷

تمرین‌ها ۶۰

فصل سوم: نام‌ها و خانواده‌های آدرس

۳-۱ نگاشت نام‌ها به اعداد ۶۴

۳-۱-۱ دسترسی به سرویس نام ۶۵

۳-۱-۲ جزئیات ۷۱

۳-۲ کدهای مستقل از آدرس ۷۳

۳-۲-۱ مشتری TCP عام ۷۵

۳-۲-۲ سرویس‌دهنده TCP عام ۷۸

۳-۲-۳ تعامل میان IPv4 و IPv6 ۸۲

۳-۳ به‌دست آوردن نام‌ها از اعداد ۸۳

تمرین‌ها ۸۵

فصل چهارم: سوکت‌های UDP

۴-۱ مشتری UDP ۸۸

۴-۲ سرویس‌دهنده UDP ۹۳

۴-۳ ارسال و دریافت داده‌ها با استفاده از سوکت‌های UDP ۹۷

۴-۴ متصل نمودن سوکت‌های UDP ۹۹

تمرین‌ها ۱۰۰

فصل پنجم: ارسال و دریافت داده‌ها

۵-۱ کدگذاری اعداد صحیح ۱۰۲

۵-۱-۱ طول اعداد صحیح ۱۰۳

۵-۱-۲ چینش بایت ۱۰۶

۵-۱-۳ اعداد علامت‌دار و توسعه علامت ۱۰۷

۵-۱-۴ کدگذاری اعداد صحیح ۱۰۹

۵-۱-۵ استفاده از سوکت‌های TCP در قالب جریان‌های فایل ۱۱۳

۱۱۶.....	۶-۱-۵ ساختارها.....
۱۲۰.....	۷-۱-۵ رشته‌ها و متن‌ها.....
۱۲۳.....	۸-۱-۵ کدگذاری مقادیر بولین.....
۱۲۵.....	۲-۵ ساخت، فریم‌بندی و تجزیه پیام‌ها.....
۱۳۴.....	۱-۲-۵ فریم‌بندی.....
۱۴۲.....	۲-۲-۵ کدگذاری مبتنی بر متن.....
۱۴۵.....	۳-۲-۵ کدگذاری دودویی پیام‌ها.....
۱۴۸.....	۴-۲-۵ کامپایل برنامه.....
۱۴۸.....	۳-۵ نتیجه‌گیری.....
۱۴۸.....	تمرین‌ها.....

فصل ششم: مباحث پیشرفته

۱۵۱.....	۱-۶ تنظیمات سوکت.....
۱۵۵.....	۲-۶ سیگنال‌ها.....
۱۶۰.....	۳-۶ ورودی و خروجی غیرقابل انسداد.....
۱۶۱.....	۱-۳-۶ سوکت‌های غیرقابل انسداد.....
۱۶۳.....	۲-۳-۶ ورودی/خروجی ناهمگام.....
۱۶۸.....	۳-۳-۶ تایم‌اوت‌ها.....
۱۷۴.....	۴-۶ چندوظیفگی.....
۱۷۴.....	۱-۴-۶ یک فرآیند به ازای هر مشتری.....
۱۸۳.....	۲-۴-۶ یک نخ به ازای هر مشتری.....
۱۸۷.....	۳-۴-۶ چندوظیفگی محدود.....
۱۸۹.....	۵-۶ تسهیم.....
۱۹۵.....	۶-۶ چندین گیرنده.....
۱۹۷.....	۱-۶-۶ پخش همگانی.....
۲۰۱.....	۲-۶-۶ چندپخششی.....
۲۰۸.....	۳-۶-۶ پخش همگانی در مقابل چندپخششی.....
۲۰۹.....	تمرین‌ها.....

فصل هفتم: سازوکار درونی

۲۱۴	۱-۷ بافر کردن داده‌ها و TCP
۲۱۸	۲-۷ خطر بن‌بست
۲۲۰	۳-۷ کارایی
۲۲۱	۴-۷ چرخه زندگی سوکت‌های TCP
۲۲۱	۱-۴-۷ اتصال
۲۲۷	۲-۴-۷ قطع یک ارتباط TCP
۲۳۲	۵-۷ تفکیک
۲۳۴	تمرین‌ها
۲۳۵	منابع

فصل ۱

سرآغاز

امروزه مردم برای تماس‌های تلفنی، تماشای تلویزیون، چت^۱، بازی با سایرین و خرید تقریباً هر نوع کالایی، از کامپیوترها استفاده می‌کنند. تخمین تعداد کامپیوترهای قابل دسترس موجود بر روی اینترنت مشکل است، اما می‌توانیم به جرئت بگوییم این تعداد به سرعت در حال رشد است و در مدت اندکی به میلیون‌ها رایانه خواهد رسید. به علاوه همه روزه برنامه‌های جدیدی ایجاد می‌شوند. با نیاز روزافزون به گسترش پهنای باند^۲ و دسترسی، اینترنت در آینده نزدیک توسعه بیشتری خواهد یافت.

اما چگونه یک برنامه با برنامه‌ای دیگر در شبکه ارتباط برقرار می‌کند؟ هدف این کتاب قراردادن شما در مسیر یافتن پاسخ این سؤال با استفاده از زبان برنامه‌نویسی C است. برای مدت زمانی طولانی، C زبان منتخب پیاده‌سازی نرم‌افزارهایی بوده است که با استفاده از شبکه با یکدیگر ارتباط برقرار می‌کردند. در واقع API^۳ که با نام سوکت^۴ شناخته می‌شود در ابتدا با استفاده از زبان C پیاده‌سازی شده است.

پیش از پرداختن به جزئیات مربوط به سوکت‌ها و چگونگی برنامه‌نویسی با استفاده از آن‌ها، بررسی کلیات شبکه‌ها و پروتکل‌ها خالی از لطف نیست. در اینجا هدفمان آموزش چگونگی کارکرد

1. Instant Messaging
2. Bandwidth
3. Application Programming Interface
4. Socket

TCP/IP یا شبکه‌ها نیست (منابع بسیار خوبی در این زمینه وجود دارند [1,3,10,15,17])، بلکه هدف اصلی‌مان، بررسی برخی از مفاهیم پایه‌ای و اصطلاحات^۱ مرتبط است.

۱-۱ شبکه‌ها، بسته‌ها^۲ و پروتکل‌ها^۳

هر شبکه کامپیوتری از تعدادی ماشین تشکیل شده است که به وسیله کانال‌های ارتباطی به یکدیگر متصل شده‌اند. این ماشین‌ها به دو گروه میزبان‌ها^۴ و مسیریاب‌ها^۵ تقسیم می‌شوند. میزبان‌ها کامپیوترهایی هستند که برنامه‌های کاربردی نظیر مرورگر وب^۶ یا برنامه‌های به اشتراک‌گذاری فایل‌ها را اجرا می‌کنند. برنامه‌های کاربردی که روی میزبان‌ها اجرا می‌شوند "کاربران" واقعی شبکه هستند. مسیریاب‌ها (که به آن‌ها دروازه^۷ نیز می‌گویند) ماشین‌هایی هستند که وظیفه ارسال اطلاعات از یک کانال ارتباطی^۸ به کانالی دیگر را بر عهده دارند. مسیریاب‌ها نیز ممکن است برنامه‌هایی را اجرا کنند اما این برنامه‌ها کاربردی نیستند. منظور از کانال ارتباطی امکان حمل دنباله‌ای از بایت‌ها از ماشینی به ماشین دیگر است، بدین منظور می‌توان از تکنولوژی‌های مبتنی بر سیم^۹ یا بی‌سیم^{۱۰} استفاده کرد.

مسیریاب‌ها از اهمیت به‌سزایی برخوردارند چرا که اتصال مستقیم یک میزبان به تمامی میزبان‌های دیگر ممکن نیست. در عوض تعدادی میزبان به یک مسیریاب متصل می‌شوند که خود به مسیریاب‌های دیگر متصل است. این روند ادامه دارد و با استفاده از آن شبکه شکل می‌گیرد. با استفاده از این روش هر ماشین می‌تواند با حداقل تعداد کانال‌های ارتباطی ممکن (عموماً یک کانال) به شبکه متصل شود. با این حال برنامه‌هایی که به تبادل اطلاعات^{۱۱} با استفاده از شبکه‌ها می‌پردازند، به صورت مستقیم با مسیریاب‌ها در ارتباط نیستند و از وجود آن‌ها اطلاعی ندارند. منظور ما از اطلاعات دنباله‌ای از بایت‌ها است که به‌وسیله برنامه‌ها ایجاد می‌شود و مورد استفاده قرار می‌گیرد. در شبکه‌های کامپیوتری به این دنباله‌های ساخته‌شده از بایت‌ها اصطلاحاً

-
1. Terminology
 2. Packets
 3. Protocols
 4. Hosts
 5. Routers
 6. Web Browser
 7. Gateway
 8. Communication Channel
 9. Wired
 10. Wireless
 11. Information

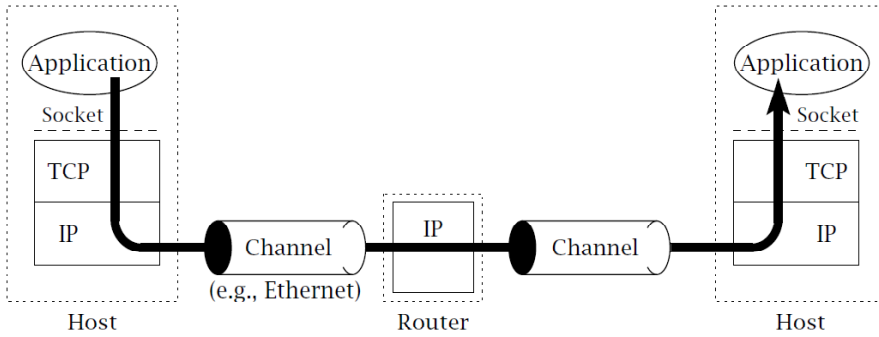
بسته می‌گویند. هر بسته حاوی اطلاعات کنترلی^۱ است که شبکه از آن‌ها به منظور انجام اعمال مربوط به خود استفاده می‌کند. برای مثال می‌توان اطلاعات مربوط به مقصد بسته را نام برد. مسیریاب‌ها از چنین اطلاعات کنترلی برای بررسی چگونگی مسیریابی هر بسته استفاده می‌کنند. پروتکل قراردادی است که نوع بسته‌های مبادله‌شده میان برنامه‌ها (منظور برنامه‌هایی است که به وسیله شبکه به مبادله اطلاعات می‌پردازند) و مفهوم هر بسته را مشخص می‌کند. در واقع پروتکل ساختار بسته‌ها - به عنوان مثال، مکان قرارگیری اطلاعات مربوط به گیرنده بسته و طول آن - و چگونگی تفسیر هر بسته را نیز مشخص می‌کند. عموماً هر پروتکل برای انجام کاری بخصوص طراحی می‌شود. برای مثال، پروتکل HTTP^۲ برای انتقال اشیای ابرمتن بین سرویس‌دهنده‌ها و مرورگرهای وب طراحی شده است که این اشیاء را در اختیار کاربر قرار می‌دهند.

پیاده‌سازی یک شبکه مفید نیازمند رسیدگی به مسائل مختلفی است. به منظور مدیریت بهتر و ماژولار^۳ نگاه‌داشتن این مسائل از پروتکل‌های متفاوتی استفاده شده است. TCP/IP یکی از این مجموعه راه‌حل‌ها است که گاهی به آن "مجموعه پروتکل"^۴ می‌گویند. TCP/IP مجموعه پروتکل‌هایی است که در اینترنت استفاده می‌شوند، البته می‌توان از آن‌ها در شبکه‌های خصوصی مستقل^۵ نیز استفاده کرد. در ادامه کتاب، هر گاه که درباره شبکه صحبت می‌کنیم، منظورمان شبکه‌هایی است که از TCP/IP استفاده می‌کنند. پروتکل‌های اصلی این مجموعه شامل پروتکل اینترنت (IP)^۶، پروتکل کنترل انتقال (TCP)^۷ و پروتکل داده‌گرام کاربر (UDP)^۸ هستند.

دسته‌بندی پروتکل‌ها به صورت لایه‌ای برای یادگیری آن‌ها بسیار رایج و مفید است. TCP/IP و تقریباً تمامی مجموعه پروتکل‌های دیگر بدین شکل سازماندهی شده‌اند. شکل ۱-۱ ارتباط میان پروتکل‌ها، برنامه‌ها و API سوکت را در میزبان‌ها و مسیریاب‌ها به همراه جریان داده‌ها از برنامه‌ای به برنامه دیگر (با استفاده از TCP) نشان می‌دهد. جعبه‌هایی که با TCP و IP عنوان‌گذاری شده‌اند پیاده‌سازی این پروتکل‌ها را نشان می‌دهند. چنین پیاده‌سازی‌هایی در سیستم عامل میزبان مقیم هستند. برنامه‌های کاربردی که از طریق API سوکت به سرویس‌هایی دسترسی دارند که به

1. Control Information
2. Hypertext Transfer Protocol
3. Modular
4. Protocol Suite
5. Standalone Private Networks
6. Internet Protocol
7. Transfer Control Protocol
8. User Datagram Protocol

وسیله TCP و UDP فراهم آورده شده‌اند با خط چین مشخص شده‌اند. پیکان‌ها جهت جریان داده را مشخص می‌کنند.



شکل ۱-۱: یک شبکه TCP/IP

در TCP/IP کانال‌های ارتباطی^۱ اصلی در پایین‌ترین لایه قرار دارند. برای مثال، در ارتباطات اترنت^۲ یا مودم‌های Dial Up: این کانال‌ها به وسیله لایه شبکه^۳ که وظیفه ارسال بسته‌ها را به مقصد بر عهده دارد (برای مثال: اعمالی که مسیریاب‌ها انجام می‌دهند) مورد استفاده قرار می‌گیرد. تنها پروتکل لایه شبکه در TCP/IP پروتکل اینترنت (IP) است.

پروتکل اینترنت سرویسی مبتنی بر داده‌گرام^۴ را فراهم می‌آورد. هر بسته در شبکه به صورت مجزا کنترل و تحویل داده می‌شود، درست مانند نامه‌ها یا بسته‌هایی که با استفاده از پست ارسال می‌شوند. بدین دلیل هر بسته IP باید حاوی آدرس گیرنده باشد، دقیقاً مانند بسته‌های پستی که باید حاوی آدرس گیرنده باشند (در ادامه درباره آدرس‌ها بیشتر صحبت خواهیم کرد). اگرچه اکثر کمپانی‌های انتقال، تحویل بسته را تضمین می‌کنند، IP پروتکلی کوشا^۵ است. یعنی IP سعی می‌کند هر بسته را تحویل دهد، اما ممکن است بسته‌ها گم شوند یا به ترتیب ارسال دریافت نشوند یا دوبار به مقصد برسند.

1. Communication Channel
2. Ethernet
3. Network Layer

۴. به یک بسته و یا واحد اطلاعاتی که حاوی اطلاعات لازم برای تشخیص مقصد باشد اصطلاحاً داده گرام می‌گویند (Datagram). -م-

5. Best-Effort

لایه بالایی لایه IP، لایه انتقال^۱ نام دارد. این لایه امکان انتخاب میان دو پروتکل را فراهم می‌آورد: TCP و UDP. هر یک از این دو پروتکل از سرویس ارائه‌شده به‌وسیله لایه IP استفاده می‌کنند، اما عمل انتقال را به روش‌های متفاوتی انجام می‌دهند؛ بدین‌سان روش‌های انتقال متفاوتی در اختیار پروتکل‌های لایه کاربردی قرار می‌گیرد تا برحسب نیازشان از یکی از این روش‌ها استفاده کنند. TCP و UDP یک کارایی مشترک دارند: آدرس‌دهی. به یاد داریم که IP بسته‌ها را به میزبان‌ها تحویل می‌دهد؛ اما برای تحویل بسته‌ها به برنامه‌های کاربردی در حال اجرا بر روی هر میزبان متصل به شبکه به مکانیزمی دقیق‌تر نیاز داریم^۲. هر دو پروتکل TCP و UDP از آدرس‌هایی به نام "شماره درگاه"^۳ برای ایجاد تمایز بین برنامه‌های مختلف موجود بر روی هر میزبان استفاده می‌کنند. این دو پروتکل را اصطلاحاً پروتکل‌های انتقال "انتها به انتها"^۴ می‌نامند، چرا که این پروتکل‌ها داده‌ها را از برنامه‌ای به برنامه دیگر انتقال می‌دهند (در حالی‌که پروتکل IP داده‌ها را از میزبانی به میزبان دیگر منتقل می‌کند).

TCP به منظور تشخیص و جبران خطاهایی نظیر گم‌شدن بسته‌ها، دریافت بسته‌های تکراری و سایر خطاهایی که ممکن است در کانال انتقال فراهم آورده شده به‌وسیله IP رخ دهند طراحی شده است. پروتکل TCP کانالی قابل اعتماد^۵ و مبتنی بر جریان بایت‌ها^۶ را فراهم می‌آورد؛ به دلیل قابل اعتماد بودن سرویس فراهم شده به‌وسیله این پروتکل برنامه‌های کاربردی استفاده‌کننده از آن نیازی به رسیدگی به خطاهای واقع در زمان انتقال ندارند. این پروتکل پروتکلی ارتباط‌گرا^۷ است، یعنی پیش از استفاده از TCP برای مخابره داده‌ها، دو برنامه ابتدا باید یک ارتباط TCP^۸ برقرار کنند که خود شامل تکمیل ارسال یک پیام دست‌تکانی^۹ میان پیاده‌سازی TCP^{۱۰} دو کامپیوتر مرتبط است. استفاده از TCP از بسیاری جهات مشابه ورودی/خروجی^{۱۱} با استفاده از فایل‌ها است. در واقع، از نظر منطقی، استفاده از TCP مانند استفاده دو برنامه از فایل‌ها است که به‌وسیله یکی

1. Transport Layer

۲. منظور نویسنده از این جمله این است که بر روی هر میزبان ممکن است ده‌ها یا صدها برنامه در حال اجرا باشد، اما پروتکل IP تنها امکان تحویل بسته‌ها به یک میزبان خاص را فراهم می‌آورد نه امکان تحویل بسته‌ها به برنامه‌ای خاص که بر روی میزبان در حال اجرا است. -م-

3. Port Number

4. End to end

5. Reliable

6. Byte-stream

7. Connectio-Oriented

8. Connection

9. Handshake

10. Implementation

11. I/O

نوشته و به‌وسیله دیگری خوانده می‌شود. برخلاف TCP، پروتکل UDP سعی در جبران خطاهایی نمی‌کند که به علت ماهیت ذاتی IP رخ می‌دهند، بلکه تنها سرویس داده‌گرام IP را توسعه می‌دهد تا به‌جای ارتباط میزبان به میزبان بتوان از آن برای ارتباطات بین برنامه‌ها استفاده کرد. بنابراین، برنامه‌هایی که از UDP استفاده می‌کنند، باید آمادگی مقابله با از دست رفتن بسته‌ها، تغییر ترتیب آن‌ها و ... را داشته باشند.

۱-۲ آدرس‌ها

پیش از پست نامه، باید آدرس گیرنده را به صورت خوانا روی بسته درج کنید. پیش از آغاز مکالمه تلفنی باید شماره بگیرید. همین‌طور پیش از این‌که برنامه‌ای بتواند با برنامه‌ای دیگر به مخابره داده‌ها بپردازد باید به نحوی مقصد داده‌ها را مشخص کند. در TCP/IP برای شناسایی هر برنامه به آدرسی دویخشی نیاز داریم. بخش اول حاوی آدرس اینترنتی است که به‌وسیله IP مورد استفاده قرار می‌گیرد و بخش دوم حاوی شماره درگاه. شماره درگاه به‌وسیله پروتکل‌های لایه انتقال به‌کار می‌روند (TCP یا UDP). آدرس‌های اینترنتی اعداد دودویی^۱ هستند و بسته به استاندارد پروتکل IP مورد استفاده دو نسخه^۲ متفاوت دارند. رایج‌ترین نسخه، نسخه^۴ (IPv4, [12]) و دیگری نسخه^۶ (IPv6, [5]) است که به‌تازگی پا به عرصه نهاده است. IP‌های نسخه^۴، ۳۲ بیت طول دارند. با این تعداد بیت می‌توان ۴ بیلیون^۳ کامپیوتر مجزا را شناسایی کرد؛ اما امروزه این رقم برای اینترنت کفایت نمی‌کند. (ممکن است این تعداد زیاد به نظر برسد، اما به دلیل شیوه‌ای که به منظور تخصیص این آدرس‌ها مورد استفاده قرار گرفته است، تعداد بسیاری از این آدرس‌ها هدر رفته‌اند. بیش از نیمی از آدرس‌های IPv4 هدر رفته‌اند). به همین دلیل IPv6 ایجاد شد. آدرس‌های IPv6، ۱۲۸ بیت طول دارند.

۱-۲-۱ نوشتن آدرس‌های IP

برای نمایش آدرس‌های IP به گونه‌ای که برای انسان قابل فهم باشد (در مقابل استفاده از آن‌ها در برنامه‌ها)، قواعد متفاوتی در IPv4 و IPv6 مورد استفاده قرار می‌گیرند. آدرس‌های IPv4 به صورت گروهی چهارتایی از اعداد صحیح که با نقطه (.) از هم جدا شده‌اند (به عنوان مثال 10.1.2.3) نوشته می‌شوند^۴. چهار قسمت موجود در این رشته، محتویات چهار بایت آدرس IP را مشخص می‌کنند، بنابراین، هر بخش حاوی عددی است در بازه صفر تا ۲۵۵.

1. Binary Number
2. Version
3. Billion

۴. به این روش نمایش اصطلاحاً Dotted-quad می‌گویند.