

## برنامه‌نویسی سوکت

- هر پروسه برای تقاضای برقراری یک ارتباط با پروسه ای دیگر روی شبکه ، یک شماره شناسایی برای خود بر می‌گزینند. به این شماره شناسایی **آدرس پورت** گفته می‌شود.

- در سرآیند بسته ای که توسط پروتکل TCP سازماندهی می‌شود، آدرس پورت پروسه فرستنده و آدرس پورت پروسه گیرنده آن درج می‌شود.

- آدرس IP، یک ماشین یکتا را در کل شبکه مشخص می‌نماید؛ شماره پورت نیز از بین پروسه های اجرا شده بر روی آن ماشین، یکی از آنها را به عنوان مبدأ (یا مقصد) تعیین می‌کند.

## برنامه‌نویسی سوکت

- آدرس IP، آدرس پورت و نوع پروتکل می‌تواند یک پروسه یکتا و واحد را بر روی هر ماشین در دنیا مشخص نماید. در ادبیات شبکه به این آدرسها، **آدرس سوکت** گفته می‌شود.

<TCP , 193.142.22.121 , 8080>

- سنگ بنای تمام برنامه های کاربردی لایه چهارم مفهومی بنام **سوکت** است که این مفهوم به منظور برقراری ارتباط برنامه های تحت شبکه و تبادل جریان داده بین پروسه ها ابداع شد.

پورت راه دور	آدرس IP راه دور	پورت محلی	آدرس IP محلی	نوع قرارداد
--------------	-----------------	-----------	--------------	-------------

## انواع سوکت

- سوکتهای نوع **استریم** که سوکتهای اتصال گرا نامیده می شوند.
- سوکتهای نوع **دیتاگرام** که سوکتهای بدون اتصال نامیده می شوند.
- سوکتهای نوع **خام** که در روترها و دیگر تجهیزات شبکه بکار می رود.

– روش ارسال برای سوکتهای نوع استریم همان روش TCP است و بنابراین داده‌ها با رعایت ترتیب و مطمئن با نظارت کافی بر خطاهای احتمالی مبادله می شوند.

– سوکتهای نوع دیتاگرام، نامطمئن است و هیچگونه تضمینی در ترتیب جریان داده‌ها وجود ندارد.

## مراحل ارتباط سوکتی

۱: یک سوکت را که مشخصه یک ارتباط است، بوجود بیاورید:

**socket (domain , type , protocol)**

۲: به سوکتی که باز کرده اید، یک آدرس پورت نسبت بدهید:

**bind (sockfd , localaddress , addresslenght)**

۳: در مرحله بعد به سیستم عامل اعلام می کنید که کارش را برای پذیرش تقاضاهای ارتباط TCP شروع نماید:

**listen (sockfd , queuesize)**

۴: نهایتاً با استفاده از تابع **accept ()** از سیستم عامل تقاضا کنید یکی از ارتباطات را در صورت وجود به برنامه شما معرفی کند:

**accept (sockfd , remoteaddress , addresslenght)**

## مراحل ارتباط سوکتی

۵: اتصال به سرویس دهنده:

`connect (sockfd , remoteaddress , addresslength)`

۶: از دستورات `sendmsg ()` و `recvmsg ()` برای مبادله داده ها استفاده نمائید.

`sendmsg (sockfd , data , datalenght , flags)`

`recvmsg (sockfd , data , datalenght , flags)`

۷: نهایتاً ارتباط را خاتمه بدهید. این کار به دو روش امکان پذیر خواهد بود:

- قطع ارتباط دو طرفه ارسال و دریافت، توسط تابع `close ()`
- قطع یکطرفه یکی از عملیات ارسال یا دریافت، توسط تابع `shutdown ()`

## زبان برنامه نویسی جاوا

- جاوا یک زبان برنامه نویسی شیء گراست که می توان گفت بطور مستقیم از C و C++ گرفته شده است و اهدافی مثل "عدم وابستگی به ماشین اجرا" که C++ در عمل نتوانست بدان دست یابد را به نحو زیبایی پیاده سازی کرده است.

- `java.net` یک بسته از بسته های جاوا است که شامل کلاسهای برای کار با شبکه، سوکتها و URL هاست. در این بسته دو نوع کلاس سوکت استریم برای برنامه نویسی شبکه تعریف شده است:

- کلاس `Socket`: کلاسی جهت برقراری ارتباط و مبادله داده در سمت مشتری است.
- کلاس `ServerSocket`: کلاسی جهت تعریف ارتباط و مبادله داده در سمت سرویس دهنده است.