

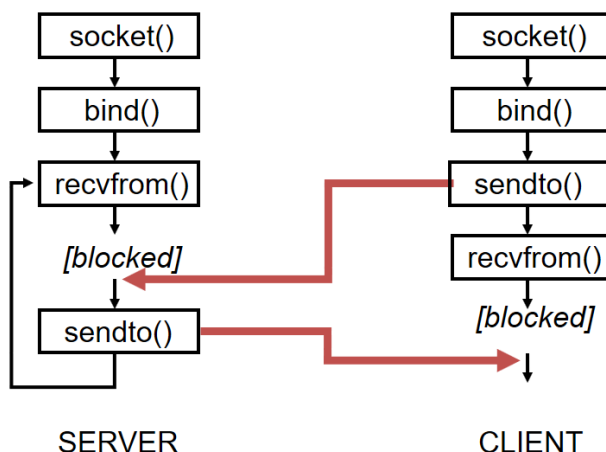
برنامه نویسی سوکت در پایتون

کلاینت و سرور

سرور کامپیوتری است که پذیرای اتصال دیگر کامپیوترها به خود می‌شود. به طور مثال در یک بازی، کامپیوتر سرور بستر ارتباط دیگر کامپیوترها به خود را با ایجاد محیط بازی و سوکت مناسب به وجود می‌آورد.

کلاینت‌ها کامپیوترهایی هستند که به سرور متصل می‌شوند.

ارتباط کلاینت و سرور به طور کلی به شکل زیر است:



بخش‌های blocked به این معنی است که دستور باعث توقف اجرای برنامه می‌شود تا اطلاعات از کامپیوتر دیگر برسد. (مثل دستور اینپوت)

سرور

```

from socket import *
server = socket(AF_INET, SOCK_STREAM)
server.bind(("192.168.88.46", 2000))
server.listen(50)

while(1):
    (cSock, cAddr) = server.accept()
    a = cSock.recv(10000)
    cSock.send(bytes('resid!', 'UTF-8'))
    print(cAddr, ': ', a)
    cSock.close()

server.close()
    
```

این برنامه را باید در سمت سرور بنویسید و اجرا کنید. توضیح دستورها را در جدول زیر مشاهده می‌کنید:

دستور	توضیح
<code>socket (AF_INET, SOCK_STREAM)</code>	تعریف سوکت (ورودی‌ها ویژگی‌های سوکت هستند و در اینجا توضیح داده نمی‌شوند)
<code>server.bind(("192.168.88.46", 2000))</code>	اتصال سوکت به آی‌پی و پورت مشخص شده
<code>server.listen(50)</code>	قرار دادن سرور در وضعیت قبول اتصال جدید (عدد توی پرانتز دلخواه است و حداکثر تعداد اتصالات غیر قابل قبول قبل از رد کردن یک اتصال را مشخص می‌کند!)
<code>server.accept()</code>	قبول یک اتصال. برنامه در این خط متوقف می‌ماند تا یک کلاینت متصل شود. خروجی آن یک دوتایی است که به ترتیب سوکت کلاینت و آدرس کلاینت را مشخص می‌کند.
<code>cSock.recv(10000)</code>	پس از اتصال اطلاعات ارسالی از سوکت کلاینت را دریافت می‌کند. عدد داخل پرانتز حداکثر حجم دریافتی را مشخص می‌کند.
<code>cSock.send(bytes('resid!', 'UTF-8'))</code>	ارسال اطلاعات به کلاینت. دستور بایتس برای تبدیل متن به اطلاعات قابل ارسال است.
<code>cSock.close()</code>	اتصال ایجاد شده با کلاینت را می‌بندد.
<code>server.close()</code>	سرور را می‌بندد و دیگر اتصالاتی را نمی‌پذیرد.

کلاینت

```
from socket import *

while (1):
    i = (input())
    mySock = socket(AF_INET, SOCK_STREAM)
    mySock.connect(("192.168.1.31", 2010))
    mySock.send(bytes(str(i), 'UTF-8'))
    print('server said:', mySock.recv(10000))
    mySock.close()
```

دستورات مربوط به کلاینت نیز همانند دستورات سرور هستند. تنها دستور جدید `connect` است که کار آن اتصال کلاینت به سوکت مشخص شده با پورت و آی‌پی است.

تمرین‌ها

- یک سرور و کلاینت بنویسید، به این صورت که کلاینت از سرور اول بودن یک عدد را بپرسد و سرور جواب دهد. (آی‌پی‌ها و پورت‌ها اولین چیزهایی هستند که باید تنظیم شوند.)
- یک سرور و کلاینت بنویسید، به این صورت که با اتصال کلاینت به سرور، سرور رمز عبور را از آن بپرسد و در صورت درست بودن پاسخ خوش آمد بگوید.
- یک چت نوبتی بین دو کامپیوتر بنویسید.
- یک چت نوبتی بین سه کامپیوتر بنویسید.
- یک چت معمولی دو نفره بین دو کامپیوتر بنویسید.

دو نمودار برای فهم بیشتر روند برنامه

