# 

# هدف آزمایش

در این دستورکار قصد داریم برنامه نویسی با سوکت در پایتون را مرور کنیم. امروزه پایتون یکی از کاربردی‌ترین زبان‌های برنامه‌نویسی است و در زمینه‌های متفاوتی می‌توان از آن استفاده کرد. در کتاب مرجع درس نیز از ویرایش ششم به بعد از زبان برنامه‌نویسی پایتون استفاده شده است.

# مقدمه

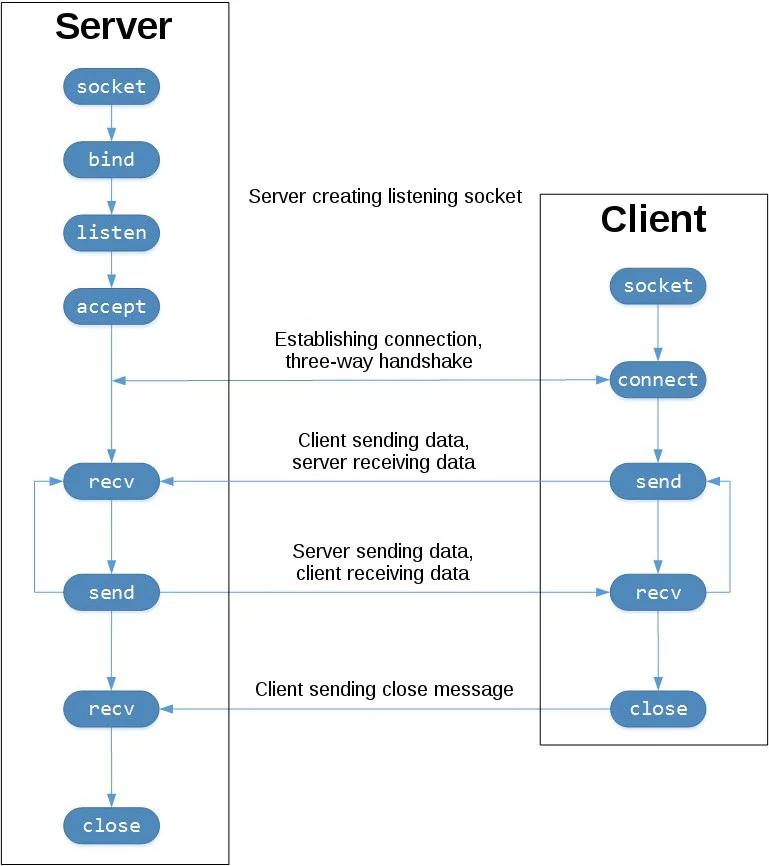
در این دستورکار قصد داریم یک سرور و کلاینت‌ با استفاده از زبان پایتون پیاده‌سازی کنیم. در این برنامه کلاینت‌ پس از متصل شدن به سرور، متن پیش فرض را به سرور ارسال می‌کند. سرور پس از دریافت متن، آن را دوباره به کلاینت‌ ارسال می‌کند. کلاینت‌ با دریافت‌ دوباره متن، متن دریافت شده را چاپ کرده و اتصال را قطع می‌کند.

# فعالیت­های قبل آزمایش

* مرور پروتکل­های لایه ارتباط
* آشنایی با یک زبان برنامه‌نویسی
* سیستم عامل لینوکس

# شرح آزمایش

برای پیاده‌سازی سرور و کلاینت Echo از پروتکل TCP برای ارتباط استفاده می‌کنیم. هدف این برنامه باز ارسال داده‌های کلاینت به همان کلاینت می‌باشد. این برنامه‌ها در گذشته برای بررسی صحت ارتباط‌های سطح بالا استفاده می‌شدند. برای یک ارتباط TCP‌ چرخه‌ی حیات زیر را می‌توان در نظر گرفت:



## سرور Echo

کد سرور به شرح زیر است‌. هر بخش کد به تفکیک در ادامه توضیح داده می‌شود.

**import** socket

HOST = '127.0.0.1' # Standard loopback interface address (localhost)

PORT = 1373 # Port to listen on (non-privileged ports are > 1023)

**with** socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) **as** s:

s.bind((HOST, PORT))

s.listen()

conn, addr = s.accept()

**with** conn:

**print**('Connected by', addr)

**while** **True**:

data = conn.recv(1024)

**if** **not** data:

**break**

conn.sendall(data)

در ابتدا یک سوکت‌ ایجاد می‌کنیم. کد زیر برای ایجاد سوکت‌ استفاده می‌شود.:

socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

ورودی اول تابع socket نوع آدرس‌دهی و ورودی دوم نوع انتقال داده است‌. در این مثال با انتخاب AF\_INET نوع آدرس دهی را از نوع IPv4 انتخاب میکنیم‌ در صورتی که قصد داریم آدرس را به صورت IP نوع چهارم یا host name استفاده کنیم باید این نوع از آدرس دهی را به عنوان ورودی به تابع بدهیم‌. SOCK\_STREAM برای ارتباط بر روی پروتکل TCP می باشد. در صورتی که قصد داشته باشیم پروتکل لایه انتقال را تغییر دهیم از این ورودی استفاده می‌کنیم.

پس از ایجاد سوکت‌ آدرس مربوطه را به آن تخصیص می‌دهیم. در اینجا آدرس داده شده به سوکت‌ آدرسی‌ است که سوکت اطلاعات را از آن دریافت و از طریق آن ارسال می‌کند. آدرس شامل دوبخش host و port است. Host یک رشته است که مشخص می‌کند این عمل گوش دادن روی کدام کارت شبکه صورت بگیرد و قسمت port مشخص کننده‌ی port گوش دهنده است. مقدار Host با توجه به نوع آدرس مشخص شده می تواند به صورت IP نوع چهارم، host name یا رشته خالی باشد. در صورتی که رشته خالی به عنوان host ارسال شود، سرور از تمام کارت های شبکه خود برای ارتباط گیری استفاده می‌کند. در نهایت با مشخص شدن این اطلاعات می‌خواهیم که سوکت در حالت قبول تقاضا قرار بگیرد.

s.bind((HOST, PORT))

s.listen(1)

پس از مشخص شدن اطلاعات آدرس، سوکت باید منتظر برقرای ارتباط بماند. برنامه پس از برخورد با کد زیر در این همان جا متوقف می شود و منتظر دریافت درخواست ایجاد یک ارتباط توسط کلاینت می ماند. پس از دریافت درخواست برقراری ارتباط، سرور با فراخوانی کد زیر درخواست را تایید می کند. پس از تایید درخواست، آدرس درخواست دهنده و ارتباط ایجاد شده در دو متغیر addr و conn ذخیر می گردد.

conn, addr = s.accept()

حالا که ارتباط برقرار شده است، امکان انتقال اطلاعات برقرار می باشد. سرور هر بار به اندازه مشخصی از داده را می خواند. در این نمونه کد در هر بار مقدار 1024 بیت داده خوانده می شود. به همین دلیل عمل خواندن داده باید در یک حلقه بی انتها قرار داده شود تا این عمل تا جایی که دیگر داده ای وجود نداشته باشد ادامه یابد.

with conn:

print('Connected by', addr)

while True:

data = conn.recv(1024)

if not data:

break

پس از اتمام دریافت داده، داده بدست آمده دوباره برای سرور ارسال می گردد. تابع sendAll تمام مقدار داده شده را برروی ارتباط برقرار شده ارسال می نماید.

conn.sendall(data)

## انجام دهید

۱. سرور Echo را مطابق با آنچه توضیح داده شد پیاده‌سازی نمایید.

۲. استفاده از مقدار

HOST = ‘0.0.0.0’

چه تغییری ایجاد می‌کند.

## کلاینت Echo به وسیله‌ی Telnet

پس از ایجاد سرور نیاز است که یک کلاینت نیز ایجاد کنیم تا بتوانیم با سرور ارتباط برقرار کنیم. یکی از راه‌های ساده تست سرور پیش از پیاده‌سازی کلاینت استفاده از دستور telnet در سیستم عامل لینوکس می‌باشد.. ساختار ساده این دستور به شکل زیر است:

telnet [ip] [port]

دستور telnet یک ارتباط TCP با آدرس IP و پورت داده شده برقرار می‌کند. شما می‌توانید هر آنچه که می‌خواهیم نوشته و با فشردن دکمه‌ی enter آن را به مقصدتان ارسال کنید.

## انجام دهید

۱. سروری که در قسمت قبل پیاده‌سازی کرده‌اید با دستور telnet آزمایش نمایید.

## کلاینت Echo

**موفق باشید.**

نسخه‌ی اولیه این دستورکار در بهار ۱۳۹۸ توسط سپهر صبور دانشجوی ارشد گرایش شبکه حاضر شده است. نسخه‌ی حاضر تنها شامل بهبودهایی اندک نسبت به نسخه‌ی اصلی می‌باشد.