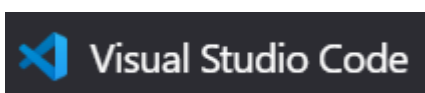


دانشگاه شهید بهشتی (دانشگاه ملی ایران)
دانشکده مهندسی و علوم کامپیوتر

درس شبکه‌های کامپیوتری ، نیم‌سال دوم سال تحصیلی ۹۸ - ۹۹
(اهنمای پروژه شماره ۱)

پروژه اول یک پروژه Socket Programming خواهد بود و با زبان پایتون نسخه ۳ انجام خواهد شد برای این منظور لازم است نسبت به نصب یکی از محیط های برنامه نویسی پایتون اقدام کنید.

دانشجویان می توانند از ابزارهای *open source* استفاده کنند ، از لینک های قرار داده شده در زیر می توانید به download ابزار متناسب با سیستم خود اقدام کنید :



[Download Page](#)



[Download Page](#)

برنامه نویسی تحت شبکه با ماژول Socket پایتون :

پایتون دو سطح دسترسی به سرویس های تحت شبکه (network services) ارائه می دهد، در سطح پایین می توانید به قابلیت های ساده ی پشتیبانی Socket در سیستم عامل دسترسی داشته باشید که به شما اجازه می دهد تا سرویس گیرنده و سرویس دهنده را برای پروتکل های اتصال گرا (Connection-oriented) و غیر اتصال گرا (connection-less) پیاده سازی کنید.

علاوه بر دسترسی سطح پایین، پایتون کتابخانه هایی دارد که دسترسی سطح بالا به پروتکل های شبکه ای سطح اپلیکیشن نظیر HTTP , FTP را فراهم می آورد.

شرح مفهوم Socket :

Socket ها در واقع End point های موجود در یک کانال ارتباطی دو طرفه هستند. Socket ها می توانند در بستر یک فرایند، بین دو فرایند در دستگاه واحد و یا چندین فرایند در دستگاه های مستقر در قاره و نقاط جغرافیایی مختلف با یکدیگر تبادل داده داشته باشند ، از دیدگاه kernel و هسته سیستم عامل، socket صرفاً نقطه ی نهایی تبادل داده و ارتباط می باشد. از دیدگاه اپلیکیشن و برنامه ی تحت شبکه، socket یک توصیف گر و شناسه ی فایل که به آن امکان و مجوز درج و خواندن داده در/ از شبکه را می دهد، قلمداد می شود.

Socket ترکیبی از آدرس دستگاه (IP) و آدرس درگاه (Port number) می باشد.

سوکت ها بر روی انواع کانال های ارتباطی قابل پیاده سازی می باشند که از جمله ی آن ها می توان به Unix domain socket، TCP، UDP و غیره ... اشاره کرد. کتابخانه ی socket کلاس های اختصاصی ارائه می دهد که علاوه بر انتقال داده های معمولی، Interface های از نوع generic که دیگر انواع عملیات انتقال و غیره را تحت پوشش قرار می دهد، مدیریت می نماید.

برای درک مفهوم سوکت و کار با آن، لازم است با واژگان زیر آشنا شوید:

واژه	شرح
Domain	خانواده ی پروتکل هایی که به عنوان مکانیزم انتقال مورد استفاده قرار میگیرد (انتقال داده در بستر شبکه بر اساس آن ها صورت می گیرد) ، این مقادیر ثوابتی همچون AF_INET و PF_INET ، PF_UNIX و PF_X25 و ... هستند.
Type	عبارت است از نوع ارتباطاتی که بین دو End Point برقرار می شود ، معمولاً SOCK_STREAM را برای پروتکل های Connection-oriented (امن و تضمین دهنده تحویل اطلاعات) و SOCK_DGRAM را ویژه پروتکل های Connection-less (نا امن با سرعت تحویل بالا که تحویل اده ها را تضمین نمی کند) را شامل می شود.
Protocol	به طور پیش فرض بر روی ۰ تنظیم می شود، این مفهوم غالباً جهت معرفی نوع دیگر از پروتکل داخل یک domain و type به کار می رود.
Host name	شناسه و نام Interface شبکه است

درس شبکه‌های کامپیوتری، نیم‌سال دوم تمصیلی ۹۸-۹۹

راهنمای پروژه شماره ۱



	<ul style="list-style-type: none"> یک رشته که می‌تواند اسم سرویس دهنده (host name)، آدرس IPV4 و IPV6 با ساختار نگارشی ۲ نقطه باشد. یک رشته "<broadcast>" که آدرس INADDR_BROADCAST را تعریف می‌کند. یک رشته با طول صفر که INADDR_ANY را تعریف می‌کند. یک عدد صحیح اختصاص داده شده به host name که معرف یک سیستم در آن شبکه است.
Port	<p>هر سرویس دهنده به کلاینت‌هایی که یک یا چند پورت را صدا می‌زنند، گوش می‌دهد، پورت می‌تواند شماره‌ی پورت Fixnum باشد، یک رشته دربردارنده‌ی شماره‌ی پورت یا اسم سرویس باشد.</p>

ماژول Socket :

به منظور ایجاد یک Socket لازم است تابع socket.socket() را فراخوانی نمایید، syntax کلی دستور به صورت زیر است :

`s = socket.socket (socket_family, socket_type, protocol=0)`

در زیر شرح هر یک از این پارامترها را مشاهده می‌کنید :

Socket_family : این پارامتر، همان طور که در بالا توضیح داده شد، می‌تواند AF_UNIX یا AF_INET باشد.

socket_type : این پارامتر می‌تواند SOCK_STREAM یا SOCK_DGRAM باشد.

Protocol : این پارامتر اختیاری بوده و به صورت پیش فرض بر روی ۰ تنظیم می‌شود.

پس از تعریف آبجکت socket، می‌توانید با استفاده از توابع لازم، برنامه‌های سمت سرویس دهنده و سمت سرویس گیرنده‌ی خود را تعریف نمایید. جداول زیر لیست توابع لازم برای این منظور را معرفی می‌کند.

Server Socket Methods (متدهای به مربوط به سمت سرویس دهنده ی از ماژول socket):

Method	شرح
s.bind()	این متد آدرس (host name یا اسم سرویس دهنده، جفت آدرس پورت یا port number pair) را به socket به صورت دوطرفه ارسال می کند.
s.listen()	این متد یک گوش فرادهنده (Listener) به TCP تنظیم و راه اندازی می کند.
s.accept()	این متد درخواست اتصال به سرویس دهنده را می پذیرد و به عبارتی ارتباط معلق را به سرور معرفی می کند.

Method های کلی ماژول Socket:

متد	شرح
s.recv()	این متد پیغام TCP را دریافت می کند.
s.send()	متد حاضر پیغام TCP را ارسال می کند.
s.recvfrom()	متد جاری پیغام UDP را دریافت می کند.
s.sendto()	این متد پیغام UDP را ارسال می کند.
s.close()	این متد socket را می بندد.
socket.gethostname()	اسم سرویس دهنده (hostname) را در خروجی برمی گرداند.

Method مازول socket مربوط به سمت سرویس گیرنده :

Method	شرح
S.connect()	اتصال به سرویس دهنده ی را بر اساس TCP راه اندازی می کند

پیاده سازی بخش مربوط به سرویس دهنده :

جهت پیاده سازی بخش مربوط به سمت سرور، تابع socket کپسوله سازی شده در مازول socket را فراخوانی کرده و ابتدا یک آبجکت socket میسازیم. سپس به واسطه ی این آبجکت، دیگر توابع لازم را جهت تنظیم و راه اندازی سرویس دهنده ی socket صدا می زنیم.

حال متد bind(hostname,port) را جهت مشخص کردن یک port برای سرویس خود در دستگاه میزبان یا سرویس دهنده فراخوانی نمایید.

سپس، متد accept() را بر روی آبجکت s (آبجکت ساخته شده ی socket) جهت معرفی ارتباط معلق به ماشین سرور فراخوانی می کنید. این متد صبر می کند که سرویس گیرنده به port یا آدرس درگاه تعیین شده، متصل شود و متعاقبا آبجکت connection را که نشانگر اتصال آن سرویس گیرنده (کلاینت) است در خروجی برمی گرداند.

```
#!/usr/bin/python      # This is server.py file
import socket          # Import socket module
s = socket.socket()     # Create a socket object
host = socket.gethostname() # Get local machine name
port = 12345           # Reserve a port for your service.
s.bind((host, port))    # Bind to the port
s.listen(5)             # Now wait for client connection.
while True:
    c, addr = s.accept() # Establish connection with client.
    print 'Got connection from', addr
    c.send('Thank you for connecting')
    c.close()           # Close the connection
```

پیاده سازی بخش سرویس گیرنده :

در این بخش از آموزش، یک اپلیکیشن ساده سمت سرویس گیرنده می نویسیم که اتصال به پورت معین ۱۲۳۴۵ و دستگاه سرویس (سرویس دهنده) را فراهم می آورد.

به راحتی می توانید یک کلاینت یا سرویس گیرنده ی socket به واسطه ی تابع مربوطه از ماژول socket ایجاد کرد. متد socket.connect(hostname, port) یک اتصال بر اساس پروتکل (TCP به hostname) دستگاه سرویس دهنده ی مربوطه بر اساس شماره ی port باز می کند. این متد بر اساس این ماشین سرور و آدرس پورت اتصال را جهت تبادل داده برقرار می نماید.

پس از باز کردن socket، می توانید اطلاعات آن را مانند سایر آبجکت های IO بخوانید. لازم است در پایان، سوکت را بسته و اتصال را خاتمه می دهید.

کد زیر همان طور که مشاهده می کنید، بخش مربوط به سرویس گیرنده و در واقع یک کلاینت ساده است که به دستگاه سرویس دهنده و شماره درگاه مربوطه وصل شده، تمامی داده های مورد نیاز را از socket می خواند و در نهایت با فراخوانی تابع close() بر روی آبجکت ، s سوکت را بسته و اتصال را خاتمه می دهد.

```
#!/usr/bin/python      # This is client.py file
import socket          # Import socket module
s = socket.socket()     # Create a socket object
host = socket.gethostname() # Get local machine name
port = 12345           # Reserve a port for your service.
s.connect((host, port))
print s.recv(1024)
s.close                # Close the socket when done
```

حال جهت مشاهده ی خروجی، ابتدا فایل server.py زیر را در پس زمینه اجرا و سپس فایل

client.py نام برده را اجرا کنید. # Following would start a server in background.

\$ python server.py &

Once server is started run client as follows:

\$ python client.py

Got connection from ('127.0.0.1', 48437)

خروجی روبرو را برمی گرداند:

Thank you for connecting

ماژول های برنامه نویسی تحت شبکه برای Python Internet / Python Modules :

در زیر لیستی از ماژول های مهم و پرکاربرد پایتون در زمینه ی برنامه نویسی تحت شبکه را مشاهده می کنید.

پروتکل	کاربرد و مورد استفاده	شماره ی درگاه یا پورت	ماژول پایتون
HTTP	Web pages (برای اپلیکیشن های تحت وب/پروتکل ارسال اطلاعات بین سرور، کلاینت)	80	httplib, urllib, xmlrpclib
NNTP	(پروتکل دسترسی به گروه های خبری)	119	nntplib
FTP	File transfers (انتقال فایل)	20	ftplib, urllib

SMTP	Sending email (ارسال ایمیل)	25	smtplib
POP3	Fetching email (واکشی ایمیل)	110	poplib
IMAP4	Fetching email (واکشی ایمیل)	143	imaplib

Telnet	Command lines (پنجره یا خط فرمان جهت اتصال به سرور راه دور)	23	telnetlib
Gopher	Document transfers ارسال فایل و سند	70	gopherlib, urllib

محمد رستمی

موفق باشید.