Socket Programming

What is Sockets?

Sockets are the endpoints of a bidirectional communications channel. Sockets may communicate within a process, between processes on the same machine, or between processes on different continents.

Sockets may be implemented over a number of different channel types: Unix domain sockets, TCP, UDP, and so on. The socket library provides specific classes for handling the common transports as well as a generic interface for handling the rest.

A Simple Server

To write Internet servers, we use the socket function available in socket module to create a socket object. A socket object is then used to call other functions to setup a socket server.

```
import socket

s = socket.socket()
```

Now call `bind(hostname, port)` function to specify a port for your service on the given host. `listen()` enables a server to accept connections.

```
s.bind(("127.0.0.1",12345))

s.listen()
```

Next, call the accept method of the returned object. This method waits until a client connects to the port you specified, and then returns a **connection object** that represents the connection to that client and the **client's address**

```
while True:
    c, addr = s.accept()  # Establish connection with client.
    print ("Got connection from", addr)
    client_msg = c.recv(1024)
    print('Recived:' + client_msg.decode())
    server_msg = "Thank you for connecting"
    c.send(server_msg.encode())
    c.close()                # Close the connection
```

Client

Let us write a very simple client program which opens a connection to a given port 12345 and given host. This is very simple to create a socket client using Python's socket module function.

The socket.connect(hosname, port) opens a TCP connection to hostname on the port. Once you have a socket open, you can read from it like any IO object. When done, remember to close it, as you would close a file.

```
import socket

s = socket.socket()

s.connect((server_ip_address, 12345))
```

```
msg = "Hello, Dr. Kim. This is abc."

s.send(msg.encode())

print(s.recv(1024))

s.close()
```

Server

```python

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(("127.0.0.1", port))        # Bind to the port

s.listen()                 # Now wait for client connection.
while True:
    c, addr = s.accept()     # Establish connection with client.
    print ("Got connection from", addr)
    client_msg = c.recv(1024)
    print('Recived:' + client_msg.decode())
    server_msg = "Thank you for connecting"
    c.send(server_msg.encode())
    c.close()                # Close the connection
```

Client

```
1
2   import socket                # Import socket module
3
4   s = socket.socket()          # Create a socket object
5   host = "127.0.0.1"           # Get local machine name
6   port = 12345                 # Reserve a port for your service.
7
8   s.connect((host, port))
9   msg = "Hello, Dr. Kim. This is abc."
10  s.send(msg.encode())
11  print(s.recv(1024))
12  s.close()                      # Close the socket when done
```

Lab 27

Develop a server application that transmits a welcome message to a client upon connection via socket. This program should also be capable of receiving a message from the client and displaying it on the console. Additionally, create a client application that establishes a connection to the server, sends a message to it, and then shows the message received from the server.

Provide a screenshot demonstrating the communication between the two programs with test messages.

Useful Links

https://www.studytonight.com/network-programming-in-python/introduction-to-network-programming