

Forum:[Hack Forums](#)

Thread: [Creating a server that speaks to clients?](#)

PGTips - July 7, 2017, 5:03 pm

I want to install a program onto 4 computers that just listens for a command by 1 main server.

How would I go about doing that?

Should I install 4 servers on each PC and my main client program speaks to all of them?

Or

Should I create a client where its just listening for calls from the server (seems more logical). If so, how would I go about doing this?

What I have so far:

client.py

Code # a simple client socket

```
import socket
```

```
# define socket address
```

```
TCP_IP = '127.0.0.1' # ip of the server we want to connect to
```

```
TCP_PORT = 5000 # port used for communicating with the server
```

```
BUFFER_SIZE = 1024 # buffer size used when receiving data
```

```
# create socket
```

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
print ("Socket created successfully.")
```

```
# connect to server
```

```
s.connect((TCP_IP, TCP_PORT))
```

```
print ("Established connection with the server.")
```

```
message = b"I've been sent from the client!"
```

```
# send message to the server
```

```
s.send(message)
```

```
print ("Message sent to server.")
```

```
# receive data from server
```

```
data = s.recv(BUFFER_SIZE)
```

```
print ("Here's the message received from the server:")
```

```
print (data)
```

Server.py

Code # a simple server socket

```
import socket
```

```
# define socket address
```

```
TCP_IP = '0.0.0.0' # consider all possible incoming IPs
```

```
TCP_PORT = 5000 # port used for communicating with the client
```

```
BUFFER_SIZE = 1024 # buffer size used when receiving data
```

```
# create socket
```

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
print ("Socket created successfully.")
```

```
# bind socket
```

```
s.bind((TCP_IP, TCP_PORT))
```

```
# start listening
```

```
s.listen(20)
```

```
print ("Waiting for a connection...")
```

```
# wait for connection
```

```
while True:
```

```
    # accept connection
```

```
    conn, addr = s.accept()
```

```
    print ("Connected with " + addr[0] + " " + str(addr[1]))
```

```
    # get message from client
```

```
    message = conn.recv(BUFFER_SIZE)
```

```
    # check that there is a message
```

```
    if not message:
```

```
        break
```

```
    # put some logic here, if needed
```

```
    server_message = b"I'm the message from the server..."
```

```
    #server_message = input("Enter message")
```

```
    #server_message = b(server_message)
```

```
    # send message to client
```

```
    conn.send(server_message)
```

```
s.close()
```

Davian - July 7, 2017, 6:32 pm

Server:

PHP Code

```
<?php
```

```
import socket
```

```
clients = []
```

```
s = socket.socket()
```

```
s.bind(('0.0.0.0', 5000))
```

```
s.listen(4)
```

```
while len(clients) != 4:
```

```
    conn, addr = s.accept()
```

```
    clients.append(conn)
```

```
while True:
```

```
    client = int(raw_input('Client 1/2/3/4: ')) - 1
```

```
    command = raw_input('Command: ')
```

```
    clients[client].send(command)
```

Client:

PHP Code

```
<?php
import socket
s = socket.socket()
s.connect(('0.0.0.0', 5000))
while True:
    command = s.recv(1024)
    if not command:
        break
    elif command == whatever:
        #Do whatever
```

nizarnizario - July 9, 2017, 8:37 am

Why don't u install Apache or Nginx in the server, and host a file that commands the clients, it's easier to deal with , you'll just need to learn some web programming language like PHP or Python (Django/Flask/Bottle...)...

PGTips - July 13, 2017, 9:53 am

(Jul 7, 2017 06:32 PM)davian Wrote: [\[CITE\]](#) Server:

PHP Code

```
<?php
import socket
clients = []
s = socket.socket()
s.bind(('0.0.0.0', 5000))
s.listen(4)
while len(clients) != 4:
    conn, addr = s.accept()
    clients.append(conn)
while True:
    client = int(raw_input('Client 1/2/3/4: ')) - 1
    command = raw_input('Command: ')
    clients[client].send(command)
```

Client:

PHP Code

```
<?php
import socket
s = socket.socket()
s.connect(('0.0.0.0', 5000))
while True:
    command = s.recv(1024)
    if not command:
        break
    elif command == whatever:
        #Do whatever
```

Bloody brilliant, I do have 1 question though.

How do I make it so that clients (if the server is up) attempts to connect to them again after X amount of time?

Sometimes the client.py will connect and after 10 minutes it will disconnect, I need them to try to

connect all the time if that makes sense?

HailPinkFloyd - July 13, 2017, 9:55 am

I love how you said "bloody brilliant" lmao

Davian - July 13, 2017, 10:04 am

(Jul 13, 2017 09:53 AM)PGTips Wrote: [\[CITE\]](#) Bloody brilliant, I do have 1 question though.

How do I make it so that clients (if the server is up) attempts to connect to them again after X amount of time?

Sometimes the client.py will connect and after 10 minutes it will disconnect, I need them to try to connect all the time if that makes sense? Just wrap client.py with while True ?

PHP Code

```
<?php
import socket
while True:
    s = socket.socket()
    s.connect(('0.0.0.0', 5000))
    while True:
        command = s.recv(1024)
        if not command:
            break
        elif command == whatever:
            #Do whatever
    s.close()
```