

## Re: Socket and cycle problem

---

*Source:* <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2008-05/msg01861.html>

---

- *From:* [petr.poupa@xxxxxxxxxx](mailto:petr.poupa@xxxxxxxxxx)
  - *Date:* Tue, 13 May 2008 10:41:43 -0700 (PDT)
- 

On 12 Kv , 21:06, Jean-Paul Calderone <exar...@xxxxxxxxxx> wrote:

On Mon, 12 May 2008 11:16:08 -0700 (PDT), petr.po...@xxxxxxxxxx wrote:

[snip]

My script send me via 3883 port (VRPN) data, but only once. I need listening this port countinously.

So I need make some loop to print data from 3883 port permanent.

Data that I recevied looks liek this:

```
receive data from server: 'vrpn: ver. 07.13 0\x00\xe8\x0b\x00\x00'
```

I'm not sure if you need to write a server or a client. In your original code, you had a client which repeatedly established out-bound connections. Here, you say you need to listen on a port.

Do you think its necessary to use Twisted? Do you have any ideas how to do it with socket modul?

The basic version is pretty easy either way. However, with Twisted, you get cross-platform error handling without any extra effort, and you don't have to think about the boring low-level details of BSD sockets.

Here's a Twisted server that listens on port 3883 forever and prints the data it receives from each connection after the remote side drops the connection:

```
from twisted.internet import reactor
```

## Re: Socket and cycle problem

```
from twisted.internet.protocol import ServerFactory, Protocol

class PrintingProtocol(Protocol):
    def connectionMade(self):
        """
        When the connection is first established, create a list
        into which to buffer all received data.
        """
        self.received = []

    def dataReceived(self, data):
        """
        Whenever any data is received on this connection, add it
        to the buffer.
        """
        self.received.append(data)

    def connectionLost(self, reason):
        """
        When the connection is lost, print out the contents of
        the receive buffer.
        """
        print repr("".join(self.received))

# Create a factory which will use our protocol to handle incoming
# connections.
factory = ServerFactory()
factory.protocol = PrintingProtocol

# Listen with it on port 3883
reactor.listenTCP(3883, factory)

# Start the reactor. Nothing in this program will ever stop the
# reactor, so it will run and accept connections forever.
reactor.run()
```

If you were to use the socket module, then it would look something like this:

```
from socket import socket
from errno import EINTR

port = socket()
port.bind(("", 3883))
port.listen(5)
while True:
    try:
        server, clientAddr = port.accept()
    except socket.error, e:
        print "Error accepting client connection", e
    else:
        received = []
```

Re: Socket and cycle problem

```
while True:
    try:
        bytes = server.recv(1024 * 16)
    except socket.error, e:
        if e.errno == EINTR:
            continue
        else:
            break
    if not bytes:
        break
    received.append(bytes)
    print repr("".join(received))
```

Hope this helps,

Jean-Paul

Thanks for code. I am trying both of them, but I am crash to another problem.

For socket modul example python report:

Traceback (most recent call last):

```
File "C:\Documents and Settings\poupa\Plocha\listen2.py", line 5, in
<module>
    port.bind(("", 3883))
File "<string>", line 1, in bind
error: (10048, 'Address already in use')
```

and for twisted example pythod report:

Traceback (most recent call last):

```
File "C:\Documents and Settings\poupa\Plocha\listen2twisted.py",
line 8, in -toplevel-
    class PrintingProtocol(Protocol):
File "C:\Documents and Settings\poupa\Plocha\listen2twisted.py",
line 28, in PrintingProtocol
    print repr("".join(self.received))
NameError: name 'self' is not defined
```

How can I solve this errors?

(for twisted I instal zope for python, pywin32 and twisted. all for python 2.4)

.