
Source: <https://foxhop.net/f3b9a5c5-2f95-11f1-beb3-e86a64d24d78/linux-nc-and-python-sockets>

Snapshot: 2026-06-15T11:53:06Z

Generator: Remarkbox 50b9d1e

This is a thread snapshot. The living document lives at the source URI above — it may have been edited, extended, or replied-to since.



Scan for living
source

linux-nc-and-python-sockets

nc or netcat lets you create socket servers or connect to services via sockets.

The nc man page is excellent. This first thing I tried was creating a simple-chat like program between two terminals.

Simple nc chat application

Start listening (nc -l) for a connection on a port (31337).

On terminal A

```
nc -l 31337
```

Connect to (127.0.0.1) on a port (31337).

On terminal B

```
nc 127.0.0.1 31337
```

Like the man page suggests all text submitted on either terminal A or B will appear on both terminals. Conventionally the server "listening" for connections is typically labeled the server. In this case neither terminal is a server because if either closes, both close...

Use python to interact with nc

On terminal A

```
nc -l 31337
```

On terminal b

```
python
```

```
>>> import socket
>>> s = socket.socket( socket.AF_INET, socket.SOCK_STREAM )
>>> s.connect( ( 'localhost', 31337 ) )
>>> s.send( 'python says hello nc' )
20
>>> s.recv( 30 )
'nc says hello python\n'
>>> # To learn more about sockets do:
>>> help( s )
>>> # when you are finished run s.close()
```

The constant `socket.AF_INET` creates a socket which allows us to connect to an ip/name and port. The constant `socket.AF_UNIX` creates a socket which allows us to connect via files? `socket.SOCK_STREAM` is the type of socket (tcp/ip). `socket.SOCK_DGRAM` (data-gram or UDP)

Source: <https://foxhop.net/f3b9a5c5-2f95-11f1-beb3-e86a64d24d78/linux-nc-and-python-sockets>

Snapshot: 2026-06-15T11:53:06Z

Generator: Remarkbox 50b9d1e