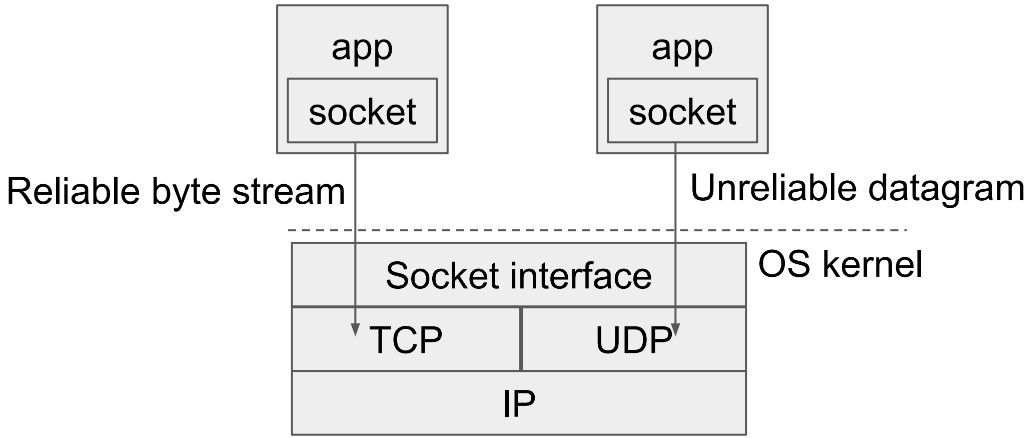
Socket Programming

Michio Honda

socket

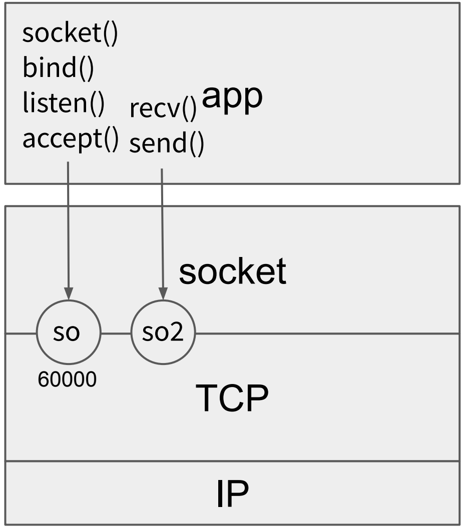
Network abstraction provided by the operating system (OS)

- Reliable byte stream
  - Like a file I/O
- Unreliable datagram
  - Datagram might be lost
  - Message boundaries



“Socket” programming in Python

```
from socket import *
so = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)
so.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1)
so.bind(('', 60000))
so.listen()
so2, _ = so.accept()
msg = so2.recv(128)
so2.send(msg)
so2.close()
so.close()
```



File (socket) descriptor

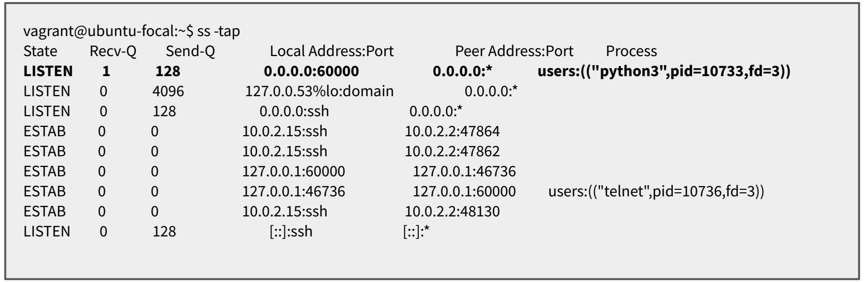
```
from socket import *
so = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)
print(so.fileno())
so.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1)
so.bind(('', 60000))
so.listen()
so2, _ = so.accept()
print(so2.fileno())
so2.send(b'hello world\n')
so2.close()
so.close()
```

Confirm you see a unique integer for each of listen and accept sockets!

Socket in the OS kernel

```
from socket import *
so = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)
print(so.fileno())
so.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1)
so.bind((", 60000))
so.listen()
while True: pass
```

(in another window)



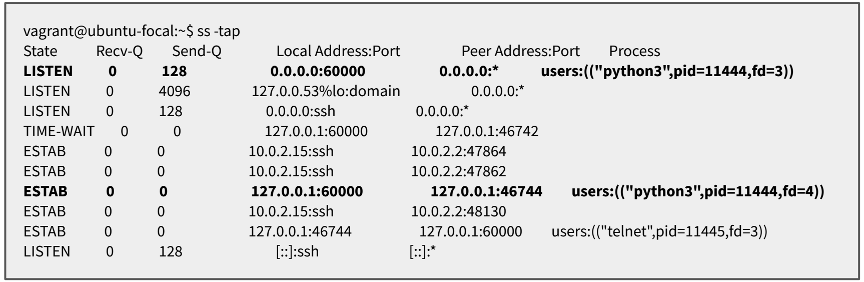
```
vagrant@ubuntu-focal:~$ ss -tap
State     Recv-Q    Send-Q      Local Address:Port          Peer Address:Port     Process
LISTEN    1         128         0.0.0.0:60000               0.0.0.0:*             users:(("python3",pid=10733,fd=3))
LISTEN    0         4096        127.0.0.53%lo:domain        0.0.0.0:*
LISTEN    0         128         0.0.0.0:ssh                 0.0.0.0:*
ESTAB     0         0           10.0.2.15:ssh               10.0.2.2:47864
ESTAB     0         0           10.0.2.15:ssh               10.0.2.2:47862
ESTAB     0         0           127.0.0.1:60000             127.0.0.1:46736
ESTAB     0         0           127.0.0.1:46736             127.0.0.1:60000       users:(("telnet",pid=10736,fd=3))
ESTAB     0         0           10.0.2.15:ssh               10.0.2.2:48130
LISTEN    0         128         [::]:ssh                    [::]:*
```

Your socket information is maintained in the kernel!

Socket in the OS kernel (2)

```
from socket import *
so = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)
print(so.fileno())
so.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1)
so.bind(('', 60000))
so.listen()
so2, _ = so.accept()
print(so2.fileno())
msg = so2.recv(128)
while True: pass
```

(in another window)



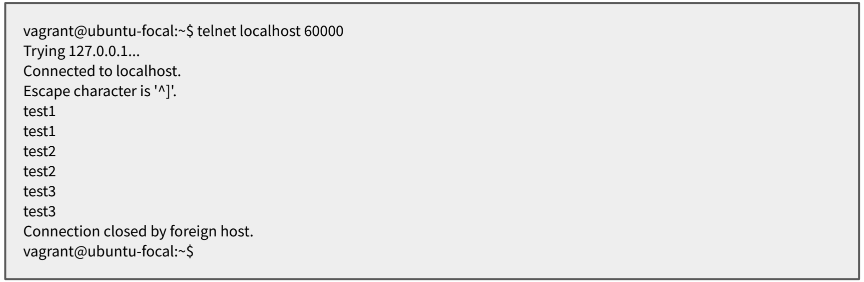
```
vagrant@ubuntu-focal:~$ ss -tap
State      Recv-Q    Send-Q      Local Address:Port        Peer Address:Port    Process
LISTEN     0         128         0.0.0.0:60000             0.0.0.0:*            users:(("python3",pid=11444,fd=3))
LISTEN     0         4096        127.0.0.53%lo:domain      0.0.0.0:*
LISTEN     0         128         0.0.0.0:ssh               0.0.0.0:*
TIME-WAIT  0         0           127.0.0.1:60000           127.0.0.1:46742
ESTAB      0         0           10.0.2.15:ssh             10.0.2.2:47864
ESTAB      0         0           10.0.2.15:ssh             10.0.2.2:47862
ESTAB      0         0           127.0.0.1:60000           127.0.0.1:46744      users:(("python3",pid=11444,fd=4))
ESTAB      0         0           10.0.2.15:ssh             10.0.2.2:48130
ESTAB      0         0           127.0.0.1:46744           127.0.0.1:60000      users:(("telnet",pid=11445,fd=3))
LISTEN     0         128         [::]:ssh                  [::]:*
```

Your socket information is maintained in the kernel!

Practice

Change this program to echo a message 3 times

(in another window)



```
vagrant@ubuntu-focal:~$ telnet localhost 60000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
test1
test1
test2
test2
test3
test3
Connection closed by foreign host.
vagrant@ubuntu-focal:~$
```