蜿蜒的大蛇

by can.

INTRO

- python twisted
- http://twistedmatrix.com
- Twisted is an **event-driven** networking engine written in Python and licensed under the open source MIT license.

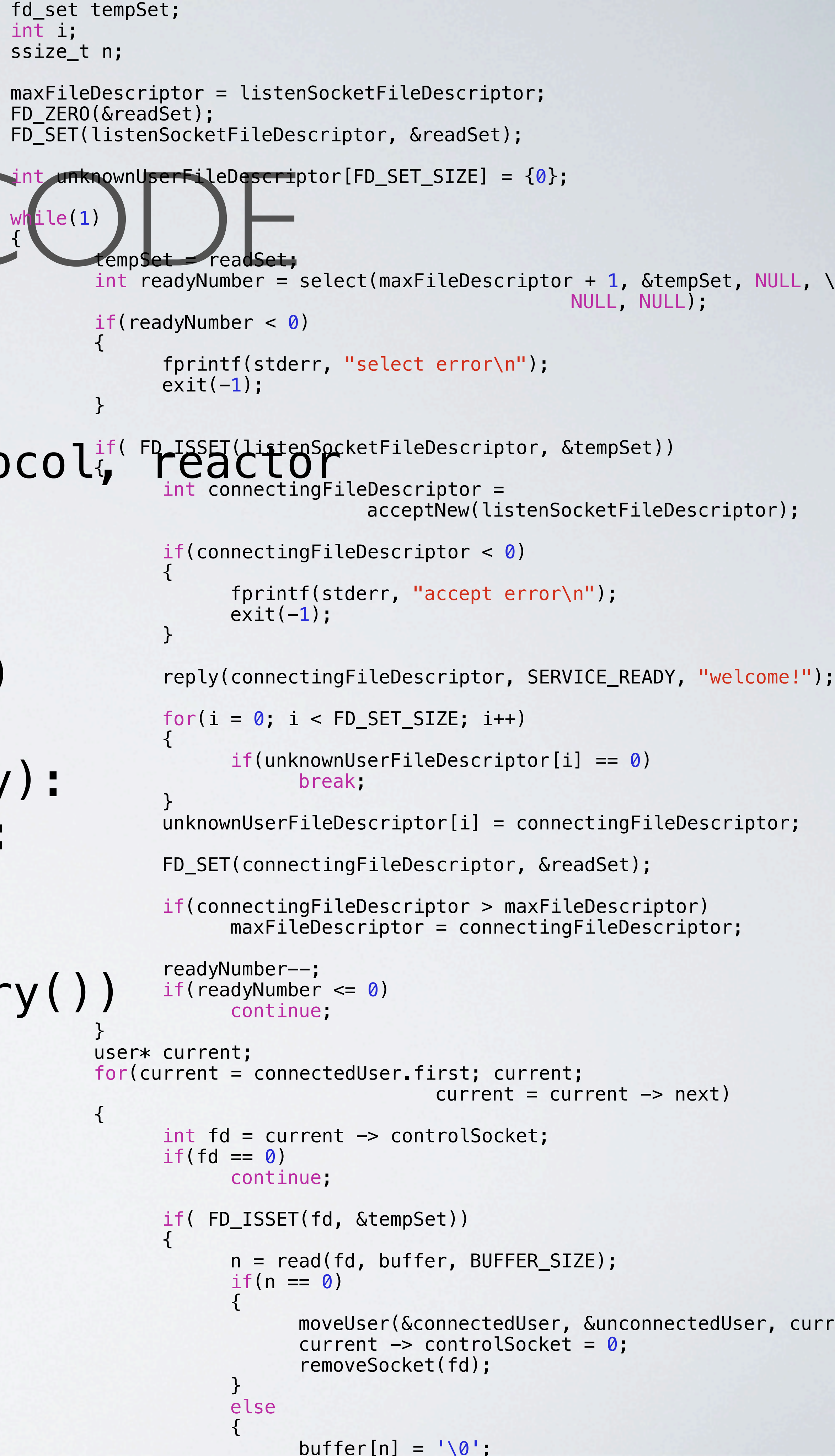
“EVENT-DRIVEN”

```
...
    def connectionMade(self):
        self.sendLine("What's your name?")

    def connectionLost(self, reason):
        if self.users.has_key(self.name):
            del self.users[self.name]

    def lineReceived(self, line):
        if self.state == "GETNAME":
            self.handle_GETNAME(line)
        else:
            self.handle_CHAT(line)
...
```

- 说白了就是回调函数

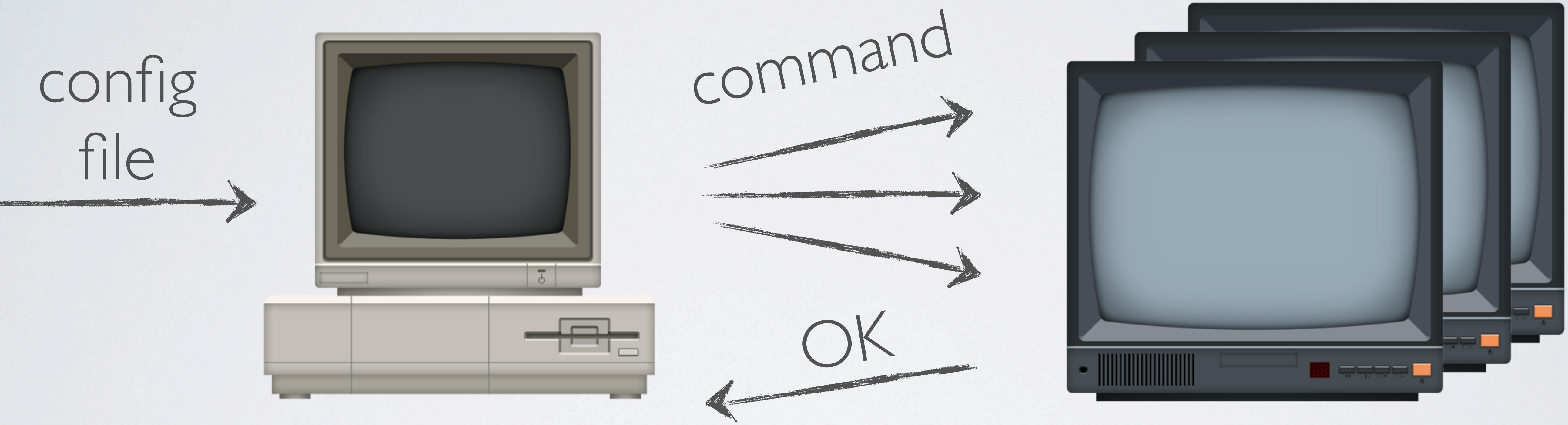
SAMPLE CODE

```python
from twisted.internet import protocol, reactor

class Echo(protocol.Protocol):
    def dataReceived(self, data):
        self.transport.write(data)

class EchoFactory(protocol.Factory):
    def buildProtocol(self, addr):
        return Echo()

reactor.listenTCP(1234, EchoFactory())
reactor.run()
```



```c
fd_set tempSet;
int i;
ssize_t n;

maxFileDescriptor = listenSocketFileDescriptor;
FD_ZERO(&readSet);
FD_SET(listenSocketFileDescriptor, &readSet);

int unknownUserFileDescriptor[FD_SET_SIZE] = {0};

while(1)
{
       tempSet = readSet;
       int readyNumber = select(maxFileDescriptor + 1, &tempSet, NULL, \
                                                  NULL, NULL);

       if(readyNumber < 0)
       {
             fprintf(stderr, "select error\n");
             exit(-1);
       }

       if( FD_ISSET(listenSocketFileDescriptor, &tempSet))
       {
             int connectingFileDescriptor =
                                acceptNew(listenSocketFileDescriptor);

             if(connectingFileDescriptor < 0)
             {
                   fprintf(stderr, "accept error\n");
                   exit(-1);
             }

             reply(connectingFileDescriptor, SERVICE_READY, "welcome!");

             for(i = 0; i < FD_SET_SIZE; i++)
             {
                   if(unknownUserFileDescriptor[i] == 0)
                         break;
             }
             unknownUserFileDescriptor[i] = connectingFileDescriptor;

             FD_SET(connectingFileDescriptor, &readSet);

             if(connectingFileDescriptor > maxFileDescriptor)
                   maxFileDescriptor = connectingFileDescriptor;

             readyNumber--;
             if(readyNumber <= 0)
                   continue;
       }
       user* current;
       for(current = connectedUser.first; current;
                                      current = current -> next)
       {
             int fd = current -> controlSocket;
             if(fd == 0)
                   continue;

             if( FD_ISSET(fd, &tempSet))
             {
                   n = read(fd, buffer, BUFFER_SIZE);
                   if(n == 0)
                   {
                         moveUser(&connectedUser, &unconnectedUser, curr
                         current -> controlSocket = 0;
                         removeSocket(fd);
                   }
                   else
                   {
                         buffer[n] = '\0';
```

PROCESS



config file

command

OK

Master (Client)

Slave (Server)