

# socket模块

# 1

socket模块

—— C/S架构

# 什么是C/S架构

- 服务器是一个软件或硬件，用于提供客户需要的“服务”
- 硬件上，客户端常见的就是平时所使用的PC机，服务器常见的有联想、DELL等厂商生产的各种系列服务器
- 软件上，服务器提供的服务主要是程序的运行，数据的发送与接收、合并、升级或其它的程序或数据的操作

# 套接字

- 套接字是一种具有“通讯端点”概念的计算机网络数据结构
- 套接字起源于20世纪70年代加利福尼亚大学伯克利分校版本的Unix
- 一种套接字是Unix套接字，其“家族名”为AF\_UNIX
- 另一种套接字是基于网络的，“家族名”为AF\_INET
- 如果把套接字比做电话的插口，那么主机与端口就像区号与电话号码的一对组合

# 面向连接与无连接

- 无论你使用哪一种地址家族，套接字的类型只有两种。一种是面向连接的套接字，另一种是无连接的套接字
- 面向连接的主要协议就是传输控制协议TCP，套接字类型为SOCK\_STREAM
- 无连接的主要协议是用户数据报协议UDP，套接字类型为SOCK\_DGRAM
- python中使用socket模块中的socket函数实现套接字的创建

# 2

socket模块

—— socket函数与方法

# 创建TCP服务器

- 创建TCP服务器的主要步骤如下：
  1. 创建服务器套接字：`s = socket.socket()`
  2. 绑定地址到套接字：`s.bind()`
  3. 启动监听：`s.listen()`
  4. 接受客户连接：`s.accept()`
  5. 与客户端通信：`recv()/send()`
  6. 关闭套接字：`s.close()`

- 编写一个TCP服务器
  1. 服务器监听在0.0.0.0的21567端口上
  2. 收到客户端数据后，将其加上时间戳后回送给客户端
  3. 如果客户端发过来的字符全是空白字符，则终止与客户端的连接



# 创建TCP客户端

- 创建TCP客户端的步骤主要如下：
  1. 创建客户端套接字：`cs = socket.socket()`
  2. 尝试连接服务器：`cs.connect()`
  3. 与服务器通信：`cs.send()/cs.recv()`
  4. 关闭客户端套接字：`cs.close()`

- 编写一个TCP客户端
  1. 连接服务器的21567
  2. 接收用户从键盘上的输入
  3. 发送接收到的字符串给服务器
  4. 如果用户按ctrl + c则退出程序

# 创建UDP服务器

- 创建UDP服务器的主要步骤如下：
  1. 创建服务器套接字：`s = socket.socket()`
  2. 绑定服务器套接字：`s.bind()`
  3. 接收、发送数据：`s.recvfrom()/ss.sendto()`
  4. 关闭套接字：`s.close()`

- 编写一个UDP服务器
  1. 服务器监听在0.0.0.0的21567端口上
  2. 收到客户端数据后，将其加上时间戳后回送给客户端

# 创建UDP客户端

- 创建UDP客户端的步骤主要如下：
  1. 创建客户端套接字：`cs = socket.socket()`
  2. 与服务器通信：`cs.sendto()/cs.recvfrom()`
  3. 关闭客户端套接字：`cs.close()`

- 编写一个UDP客户端
  1. 连接服务器的21567
  2. 接收用户从键盘上的输入
  3. 发送接收到的字符串给服务器
  4. 如果用户按ctrl + c则退出程序



更多精彩...



<http://bj.linux.tedu.cn/>  
企业QQ：86198501