

## TCP 客户端服务器编程模型

### 1 客户端调用序列

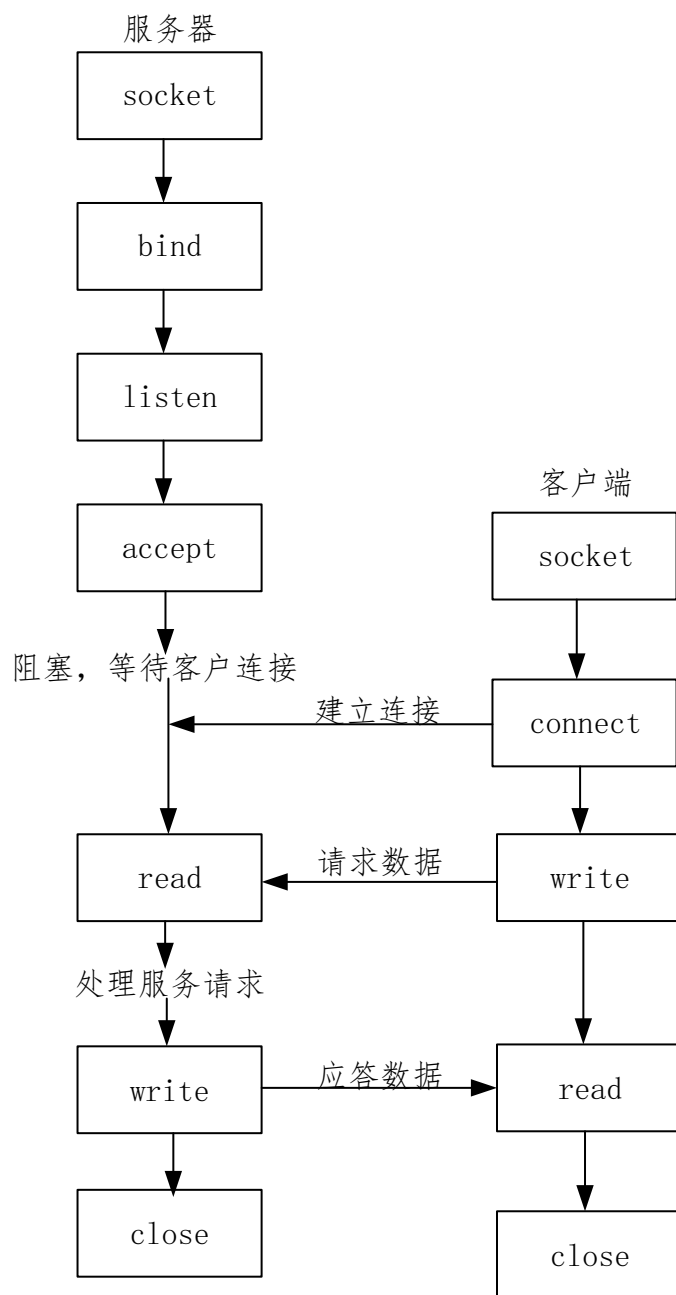
客户端编程序列如下：

1. 调用 socket 函数创建套接字
2. 调用 connect 连接服务器端
3. 调用 I/O 函数（read/write）与服务器端通讯
4. 调用 close 关闭套接字

### 2 服务器端调用序列

服务端的编程序列如下：

1. 调用 socket 函数创建套接字
2. 调用 bind 绑定本地地址和端口
3. 调用 listen 启动监听
4. 调用 accept 从已连接队列中提取客户端连接
5. 调用 I/O 函数（read/write）与客户端通讯
6. 调用 close 函数关闭套接字



### 3 常用函数

#### 3.1 套接字与地址绑定函数

##### (1.) 绑定地址

```
#include <sys/socket.h>
```

```
int bind(int sockfd, const struct sockaddr *addr, socklen_t len);
```

返回：成功返回 0，出错返回 -1

## 特殊 bind 地址介绍

一个主机可以有多个网络接口和多个 IP 地址，如果我们只关心某个地址的连接请求，我们可以指定一个具体的本地 IP 地址，如果要响应所有接口上的连接请求就要使用一个特殊的地址 INADDR\_ANY

```
#define INADDR_ANY    (uint32_t)0x00000000
```

监听所有服务器上 IP 得到的连接请求

```
struct sockaddr_in servaddr;  
  
memset(&servaddr, 0, sizeof(servaddr));  
  
servaddr.sin_addr.s_addr = INADDR_ANY;
```

## 3.2 查找绑定到套接字的地址

```
#include <sys/socket.h>  
  
int  getsockname(int  sockfd,  struct  sockaddr  *restrict_addr,  socklen_t  
*restrict_alenp);
```

返回：成功返回 0，出错返回 -1

## 3.3 获取对方地址

```
int  getpeername(int  sockfd,  struct  sockaddr  *restrict_addr,  socklen_t  
*restrict_alenp);
```

返回：成功返回 0，出错返回 -1

## 3.4 服务器端建立连接

```
#include <sys/socket.h>  
  
int listen(int sockfd, int backlog);
```

返回：成功返回 0，出错返回 -1。

backlog 指定进行客户端连接排队的队列长度。

```
int accept(int sockfd, struct sockaddr *restrict_addr, socklen_t *restrict_len);
```

返回：成功返回一个新的 sockfd（客户端的）。

### 3.5 客户端请求连接

```
#include <sys/socket.h>
```

```
int connect(int sockfd, const struct sockaddr *addr, socklen_t len);
```

返回：成功返回 0，出错返回 -1