

第二章 套接字类型与协议设置

协议（Protocol）是为了完成数据交换而定好的约定。

创建套接字（Linux版本）

```
#include <sys/socket.h>
int sock_fd = socket(int domain, int type, int protocol); //成功时返回文件描述符，失败
时返回-1
//domain: 套接字中使用的协议族（Protocol Family）信息
//type: 套接字数据传输类型信息
//protocol: 计算机间通信中使用的协议信息
```

协议族（Protocol Family）

套接字中实际采用的最终协议信息是通过socket函数的第三个参数传递的，在指定的协议族范围内通过第一个参数决定第三个参数。

表2-1 头文件sys/socket.h中声明的协议族

名 称	协 议 族
PF_INET	IPv4互联网协议族
PF_INET6	IPv6互联网协议族
PF_LOCAL	本地通信的UNIX协议族
PF_PACKET	底层套接字的协议族
PF_IPX	IPX Novell协议族

套接字类型之 SOCK_STREAM、SOCK_DGRAM

套接字的数据传递方式，对应socket函数的第二个形参(type)。

SOCK_STREAM（面向连接的套接字）：可靠的、按序传递的、基于字节的面向连接的数据传输方式的套接字

- 传输过程中数据不会消失。
- 按顺序传输数据（类似于传送带）。
- 传输的数据不存在数据边界（由于收发数据的套接字内部有缓冲区(buff), 简言之就是字节数组。通过套接字传输的数据将保存到该数组。因此，收到数据并不意味着马上调用read函数。只要不超过数组容量，则有可能在数据填满缓冲区后通过一次read函数调用读取全部，也有可能分成多次read函数调用进行读取。也就是说，在面向连接的套接字中，read函数和write函数的调用次数并无太大意义。所以说面向连接的套接字不存在数据边界）。

tips: 1、在使用write时，第三个参数使用strlen()更好，不需要减一。
2、套接字缓冲已满并不意味着数据的丢失，因为当数据缓冲区已满时，write函数会暂停输入，直到有read读取后留出空余空间write函数继续执行。

SOCK_DGRAM（面向消息的套接字）：不可靠的、不按序传递的、以数据的高速传输为目的的套接字

- 强调快速传输而非传输顺序
- 传输的数据可能丢失也可能损毁
- 传输的数据有数据边界（接收数据的次数应和传输次数相同）
- 限制每次传输的数据大小

创建套接字（Windows版本）

```
#include <winsock2.h>
SOCKET hServSock = socket(int af, int type, int protocol);
//成功时返回socket句柄，失败时返回INVALID_SOCKET
```

第二章习题

1、什么是协议？在收发数据中定义协议有何意义？

答：协议是指数据传递过程中需要遵守的约定，只有遵循一定的约定（协议），才能进行不同形式的收发数据方式（TCP、UDP）。

2、面向连接的TCP套接字传输特性有3点，请分别说明。

答：1、面向连接——消息的传递前提是客户端与服务端要连接正常。

2、可靠的——传输的数据会保证完整性，不会造成数据丢失。

3、无消息边界——数据套接字有个数据缓冲区，write函数与read函数并没有直接的关联，只与该缓冲区有直接关系；多次write函数的执行可能只对应一次read函数调用，简单来说一次read函数的调用就可以收取多次write函数调用所发送的数据。

3、下列哪些是面向消息的套接字的特性？

a、传输数据可能丢失

b、没有数据边界(Boundary)

c、以快速传递为目标

d、不限制每次传递数据的大小

e、与面向连接的套接字不同，不存在连接的概念

答：a、c、e。

4、下列数据适合用哪类套接字传输？并给出原因。

a、演唱会现场直播的多媒体数据

答：演唱会直播适合UDP传输，其优点是UDP不等待确认，也不重传丢失的数据，可以减少传输延迟，保证了实时性；其次UDP协议简单，传输开销低，可以在有限的带宽下传输更多的数据，能更高效地利用网络资源。

b、某人压缩过的文本文件

答：适合使用TCP传输，文本文件需要保证其完整性，TCP不仅可以确保传输文件的完整性，也可以保证数据的有序性，并且TCP内置了错误检测和自动重传机制，进一步的注重传输的数据完整性。而UDP是不可靠的，并且不会重传，不能保证传输数据的完整性和有效性。

c、网上银行用户与银行之间的数据传递

答：适合使用TCP协议的套接字传输，银行首要考虑的是账户的安全性，那么需要的是一个面向连接、可靠的协议，这样才能保证数据传递的完整性和可靠性。银行交易数据传输优先考虑可靠性和安全性，而TCP的特点完全满足这些需求。

5、何种类型的套接字不存在数据边界？这类套接字接收数据时需要注意什么？

答：面向字节流的套接字（如TCP套接字）不存在数据边界，这意味着发送的数据被看作是一个连续的字节流，没有明确的消息或数据包的边界。接收这种套接字的数据时，需要注意以下几点：

1、数据可能分段到达：发送端一次发送的数据可能会被网络层分成多个小块到达接收端，因此需要多次调用recv()函数来完整接收一条逻辑消息。

2、数据可能合并到达：网络可能将多次发送的小数据块合并到一起发送到接收端，因此需要解析接收到的数据以识别逻辑消息的边界。

3、明确协议格式：由于字节流没有边界，应用层必须设计一种协议来划分消息，例如使用固定长度的消息头、特殊的分隔符、或在消息头中指明消息体长度。

4、循环接收：接收数据时需要循环读取，直到满足应用层协议所定义的逻辑消息完整性。

总体来说，在使用面向字节流的套接字时，接收端必须依靠上层协议的设计来解析数据，确保正确处理分段和合并问题。这是面向字节流的套接字与面向数据包的套接字（如UDP）处理方式的主要区别。