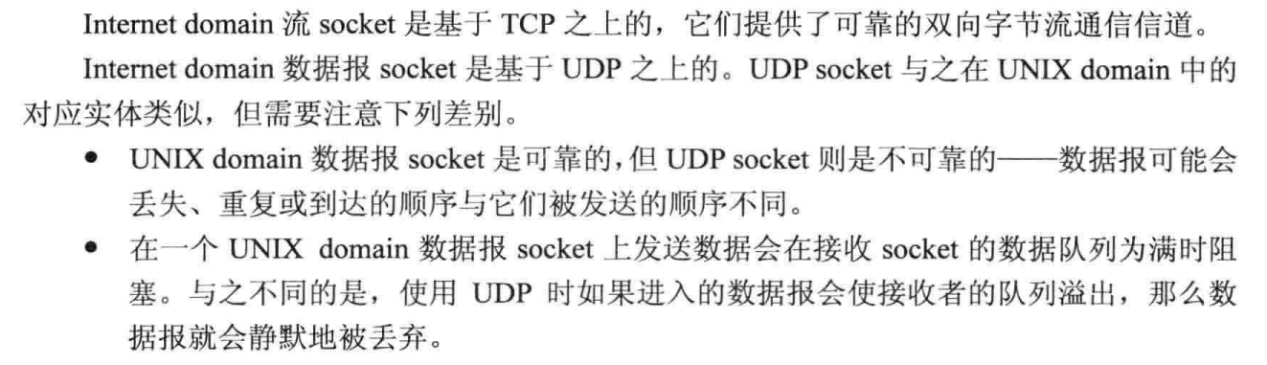
第五十九章

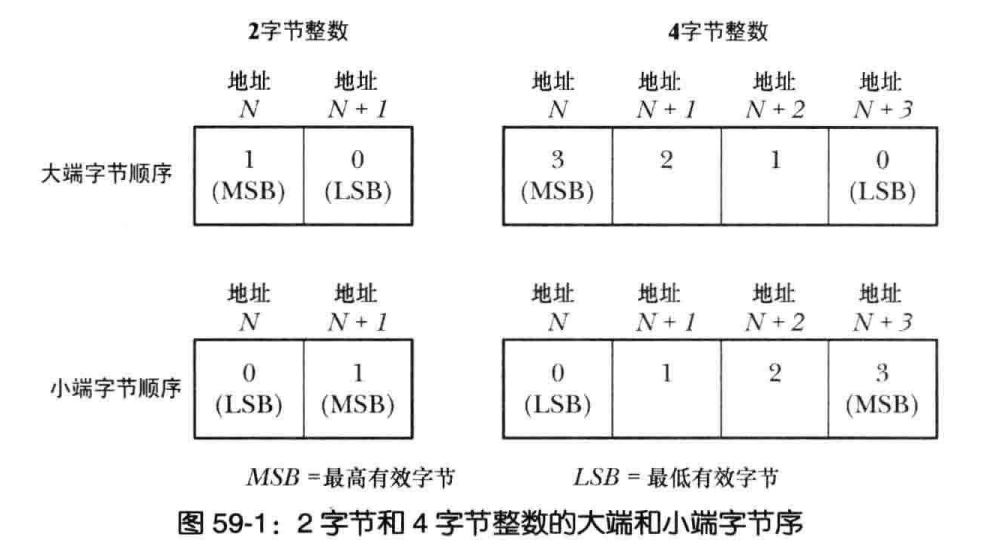
1. SOCKET：Internet Domain：
2. Internet domain socket：



1. 网络字节序：

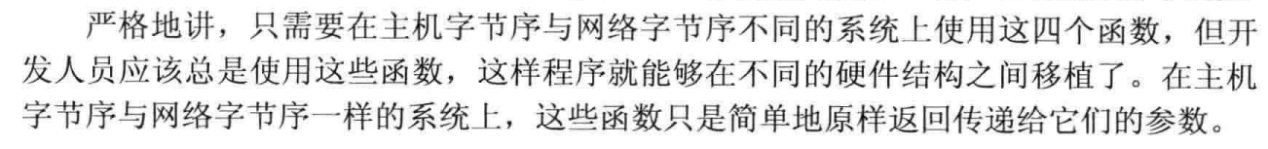
大端字节序：存储整数的时候先存储（即在最小内存地址处）最高有效位被称为大端。

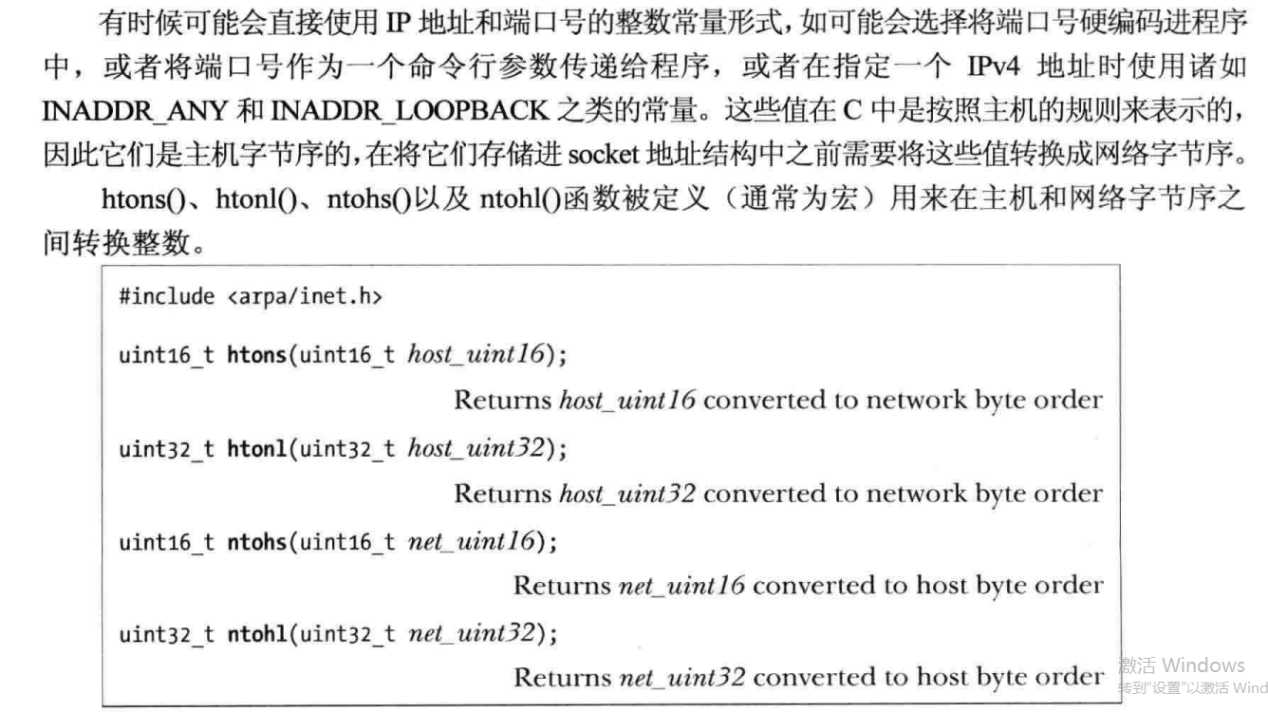
小端字节序：先存储最低有效位的被称为小端。



主机字节序：在特定主机上使用的字节序。

网络字节序是大端的。

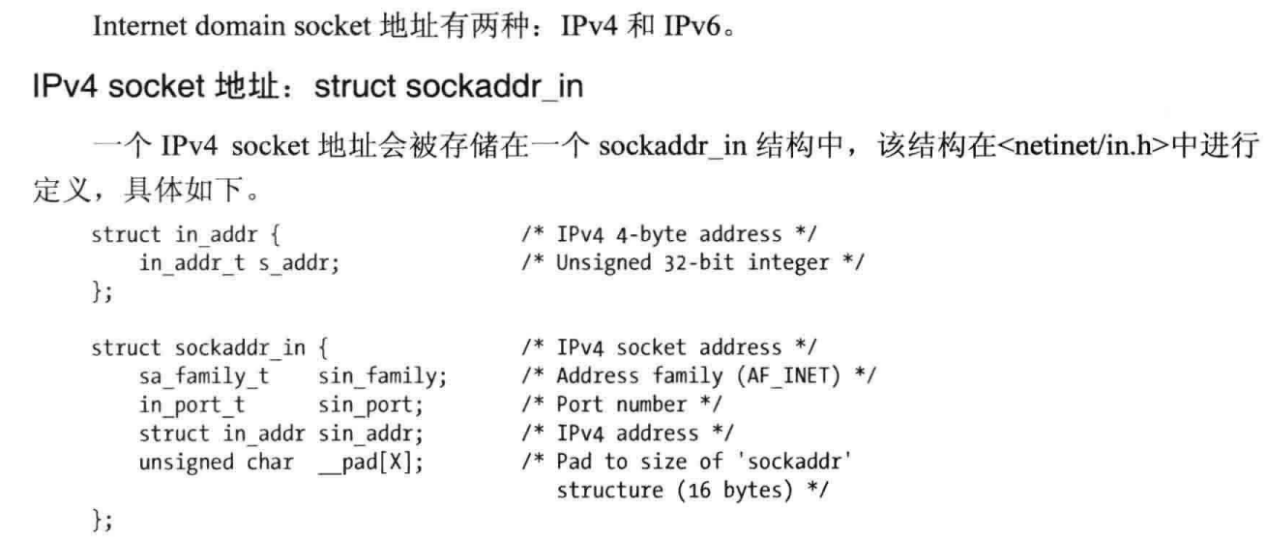


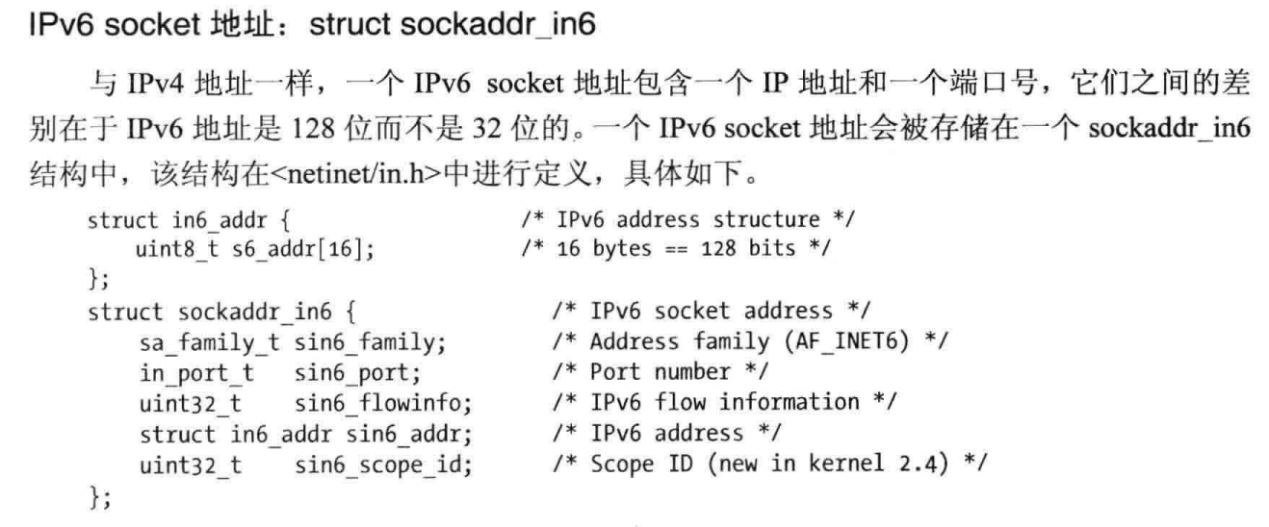


数据表示：

由于数据在表现上存在差异，因此在网络上的异构系统之间交换数据的应用程序必须要采用一些公共规则来编码数据。

1. Internet socket 地址：





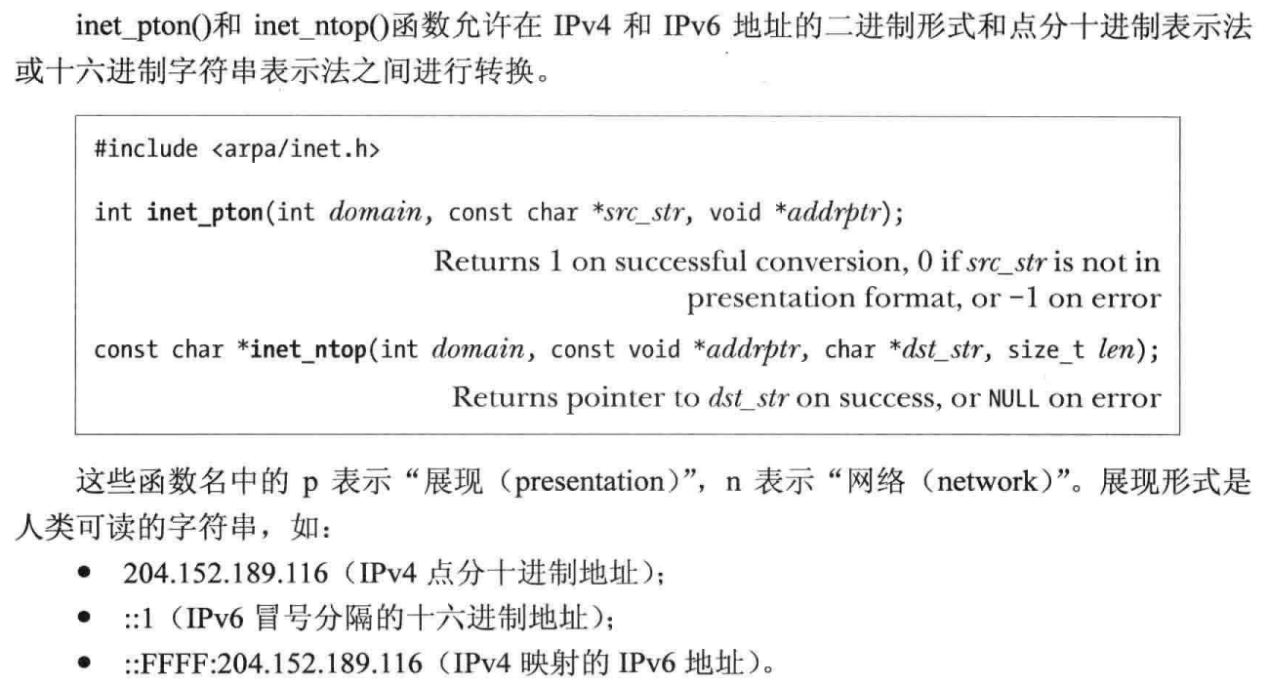
1. 主机和服务转换概述：

分别有：

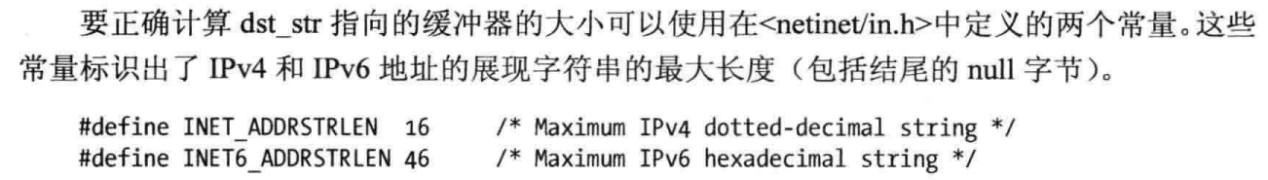
在二进制和人类可读的形式之间进行转换。

主机名和服务名与二进制形式进行转换。

1. 函数：



注意，转换后为网络字节序，不需要再次使用htons之类的函数。



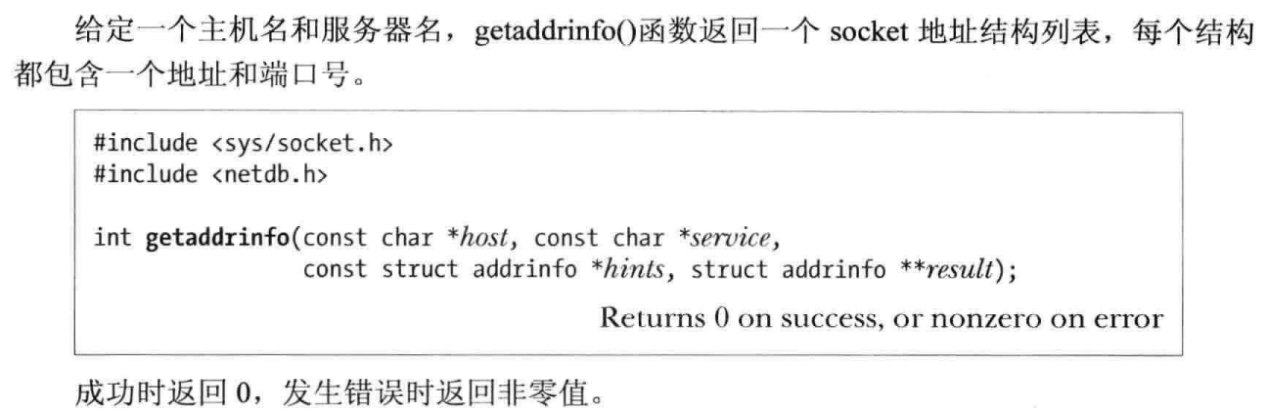
1. 域名系统：

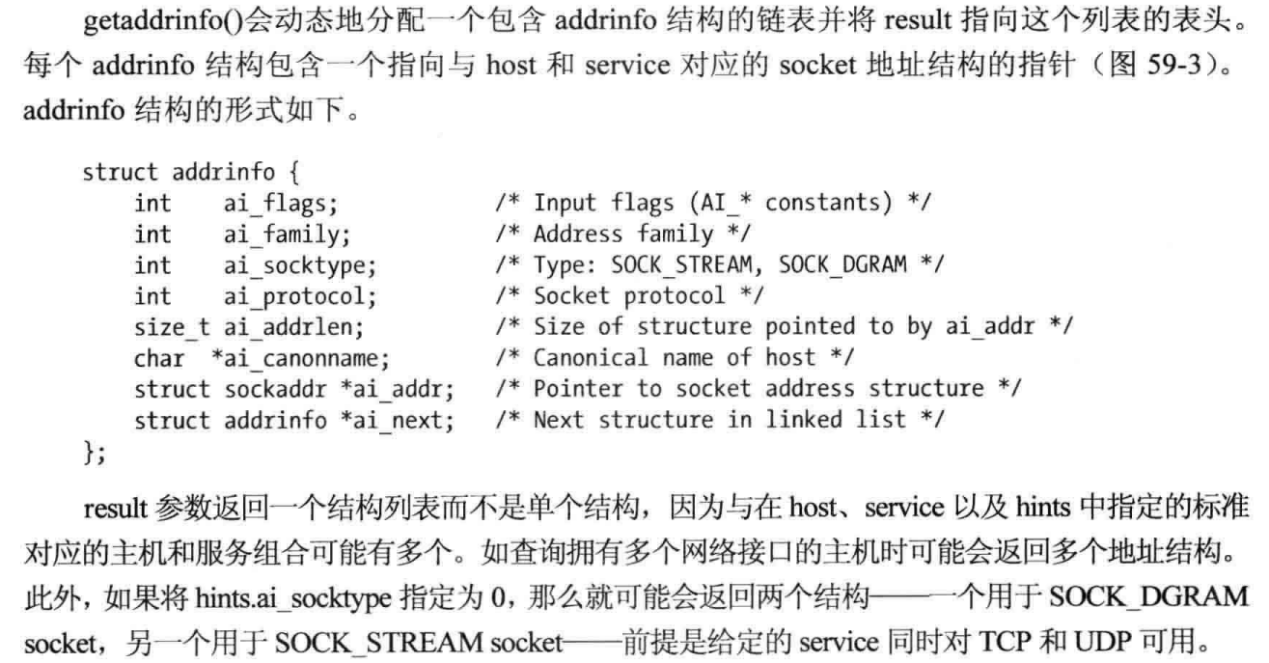
简单解释了域名系统的设计。

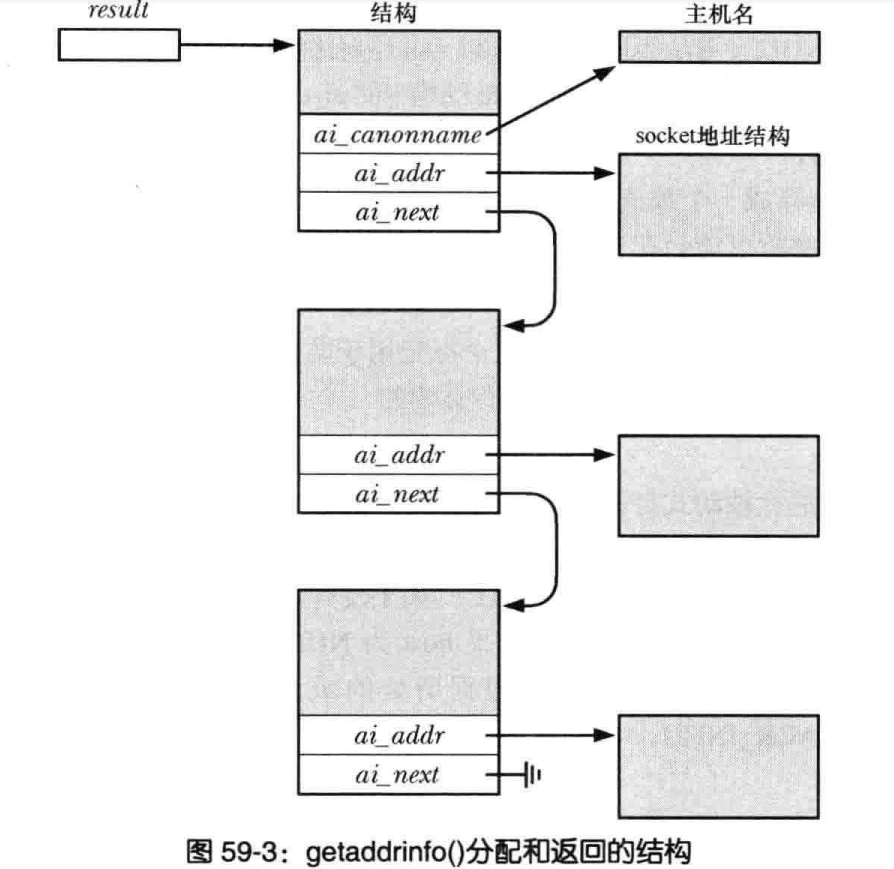
1. /etc/services 文件：

记录着端口号和服务名。

1. 独立于协议的主机和服务转换：

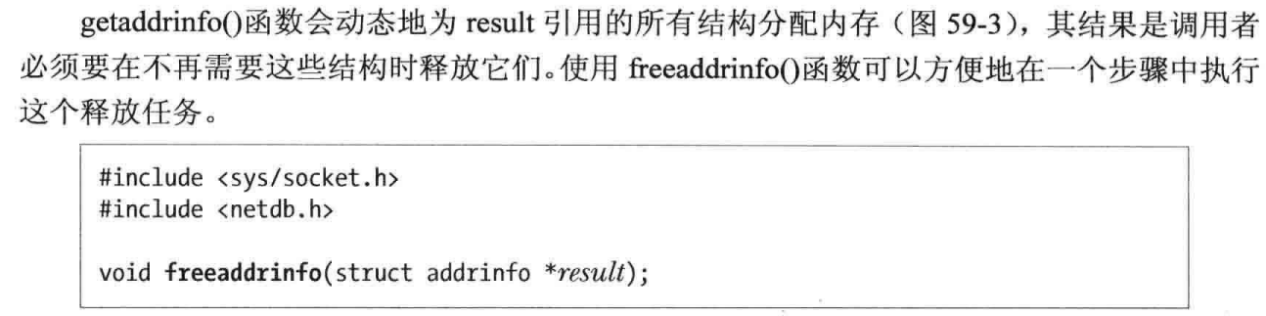




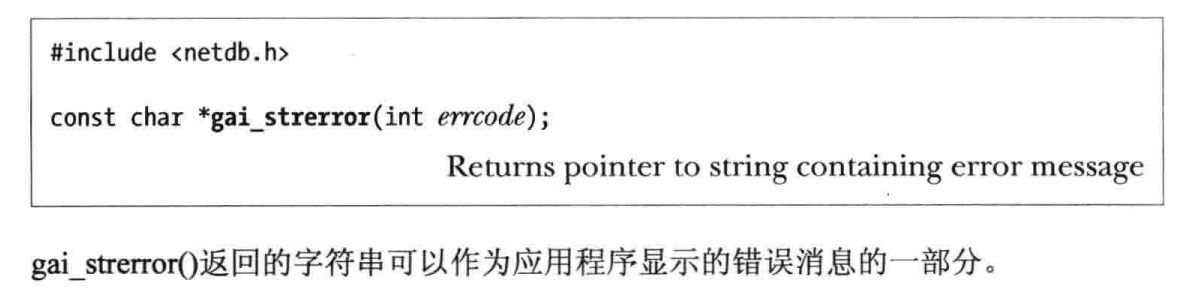


注意，hints参数可以参看手册和书本。

1. 释放列表：

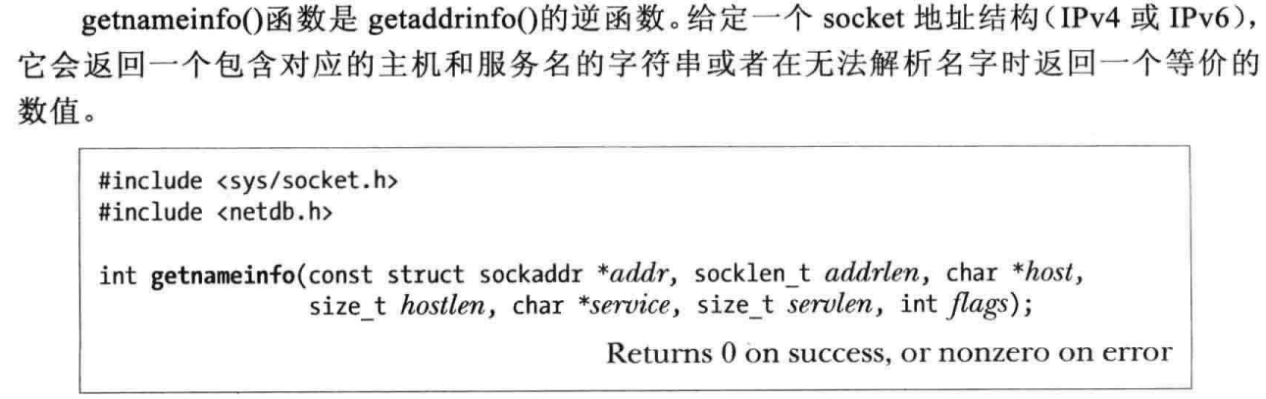


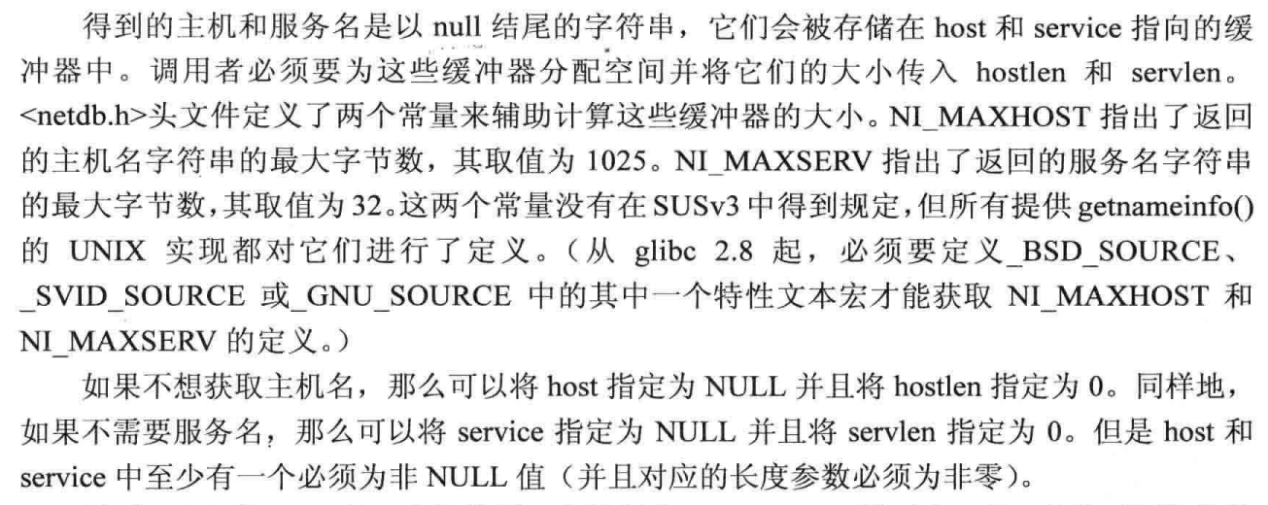
1. 错误诊断：



具体错误码参考书本或手册。

1. 函数getnameinfo：





最后一个参数flag具体含义查看书本或手册。

1. 客户端和服务端：

下图含义不太清楚：



1. Internet Socket Domain库：

实现的时候发现两个问题：

1. accept之类的函数的addrlen和addr的结构取决于服务器的还是客户端的。由此引申出来的问题是IPv4和IPv6之间能否互通？
2. 库listen函数和库bind函数共同实现使用AI\_PASSIVE的标识是否有影响，因为bind函数也有针对客户端的情况。

对于第一个问题，我尝试了一下，IPv6的服务器和IPv4的客户端可以互通，结构地址取决于服务器。反过来则出现错误，不知道是我编码问题还是什么原因。

1. 过时的主机和服务转换API：

具体查看书本和手册。

1. UNIX和Internet domain socket 的比较：

Internet domain socket 既能运行在同一主机上，也能运行在网络中不同主机上。

Unix domain的好处：

