
网络操作基础知识

一、IP 地址和端口号

1) IP 地址用于在网络中唯一标识一台机器 (通信实体), 是一个 32 位整数, 通常用 4 个 0-255 的十进制数标识;

2) 端口号用于唯一标识通信实体上进行网络通讯的程序, 同一台机器上不能有两个程序占用同一个端口的情况; 端口号是一个 16 位整数, 共有 3 类;

1. 公认端口 : 0-1023 , 它们紧密绑定一些特定的服务 (比如 TCP/IP 通信协议程序的端口);
2. 注册端口 : 1024-49151 , 它们松散绑定一些特定服务, 其余可以随意使用, 应用程序用的最多端口号来自于这个范围;
3. 动态/私有端口 : 49152-65535 , 这些端口一般是应用程序运行时临时地、动态地占用, 可以说是一种竞争资源吧, 应用程序本身不会和这些端口绑定, 一般会将其作为资源利用;

3) 真正参与网络通信的对象是计算机上的程序 (网络通信程序), IP 地址是它的街道门牌, 而端口是它的房间号;

二、阻塞 (blocking) 与非阻塞 (non-blocking) IO

IO 的阻塞、非阻塞主要表现在一个 IO 操作过程中, 如果有些操作很慢, 比如读操作时需要准备数据, 那么当前 IO 进程是否等待操作完成, 还是得知暂时不能操作后先去做别的事情? 一直等待下去, 什么事也不做直到完成, 这就是阻塞。抽空做些别的事情, 这是非阻塞。

非阻塞 IO 会在发出 IO 请求后立即得到回应, 即使数据包没有准备好, 也会返回一个错误标识, 使得操作进程不会阻塞在那里。操作进程会通过多次请求的方式直到数

据准备好，返回成功的标识。

想象一下下面两种场景：

A 小明和小刚两个人都很耿直内向，一天小明来找小刚借书：“小刚啊，你那本 XXX 借我看看”。于是小刚就去找书，小明就等着，找了半天找到了，把书给了小明。

B 小明和小刚两个人都很活泼外向，一天小明来找小刚借书：“嘿小刚，你那本 XXX 借我看看”。小刚说：“我得找一会”，小明就去打球去了。过会又来，这次书找到了，把书给了小明。

结论：A 是阻塞的，B 是非阻塞的。

从 CPU 角度可以看出非阻塞明显提高了 CPU 的利用率，进程不会一直在那等待。但是同样也带来了线程切换的增加。增加的 CPU 使用时间能不能补偿系统的切换成本需要好好评估。

三、同步 (synchronous) 与异步 (asynchronous) IO

判断同步和异步的标准在于：一个 IO 操作直到完成，是否导致程序进程的阻塞。

如果阻塞就是同步的，没有阻塞就是异步的。这里的 IO 操作指的是真实的 IO 操作，也就是数据从内核拷贝到系统进程（读）的过程。继续前面借书的例子，异步借书是这样的：

C 小明很懒，一天小明来找小刚借书：“嘿小刚，你那本 XXX 借我看看”。小刚说：“我得找一会”，小明就出去打球了并且让小刚如果找到了就把书拿给他。小刚是个负责任的人，找到了书送到了小明手上。

A 和 B 的借书方式都是同步的，有人要问了 B 不是非阻塞嘛，怎么还是同步？前面说了 IO 操作的 2 个步骤：准备数据和把数据从内核中拷贝到程序进程。映射到这个例子，书即是准备的数据，小刚是内核，小明是程序进程，小刚把书给小明这是拷贝数

据。在 B 方式中，小刚找书这段时间小明的确是没闲着，该干嘛干嘛，但是小刚找到书把书给小明的这个过程也就是拷贝数据这个步骤，小明还是得乖乖的回来候着小刚把书递手上。所以这里就阻塞了，根据上面的定义，所以是同步。

在涉及到 IO 处理时通常都会遇到一个是同步还是异步的处理方式的选择问题。

同步能够保证程序的可靠性，而异步可以提升程序的性能。小明自己去取书不管等着不等着迟早拿到书，指望小刚找到了送来，万一小刚忘了或者有急事忙别的了，那书就没了。