**Process - the smallest unit of resource allocation,**

**Thread - the smallest unit of execution of a program."**

Linux下的TCP/IP网络编程:

1. TCP/IP协议（如连接的建立和终止、重传和确认、滑动窗口和拥塞控制等等）

2. Socket I/O系统调用（重点如read/write），这是TCP/IP协议在应用层表现出来的行为。

3. 编写Performant, Scalable的服务器程序。包括多线程、IO Multiplexing、非阻塞、异步等各种技术。

**一. read/write的语义：为什么会阻塞？**

先从write说起：

#include <unistd.h>  
ssize\_t write(int fd, const void \*buf, size\_t count);

首先，write成功返回，**只是buf中的数据被复制到了kernel中的TCP发送缓冲区。**至于数据什么时候被发往网络，什么时候被对方主机接收，什么时候被对方进程读取，系统调用层面不会给予任何保证和通知。

**二. blocking（默认）和nonblock模式下read/write行为的区别：**

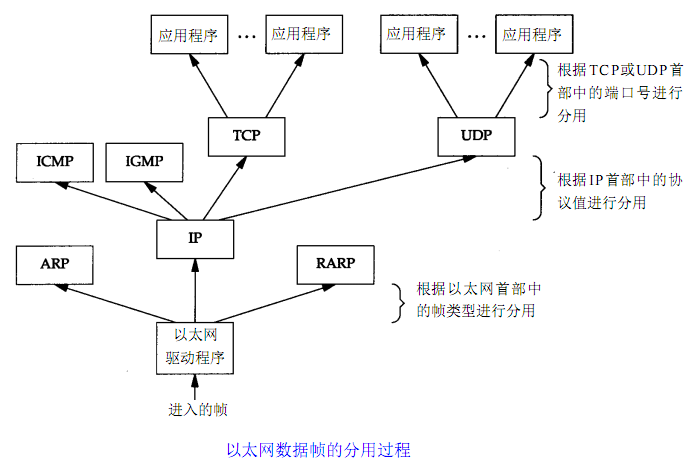
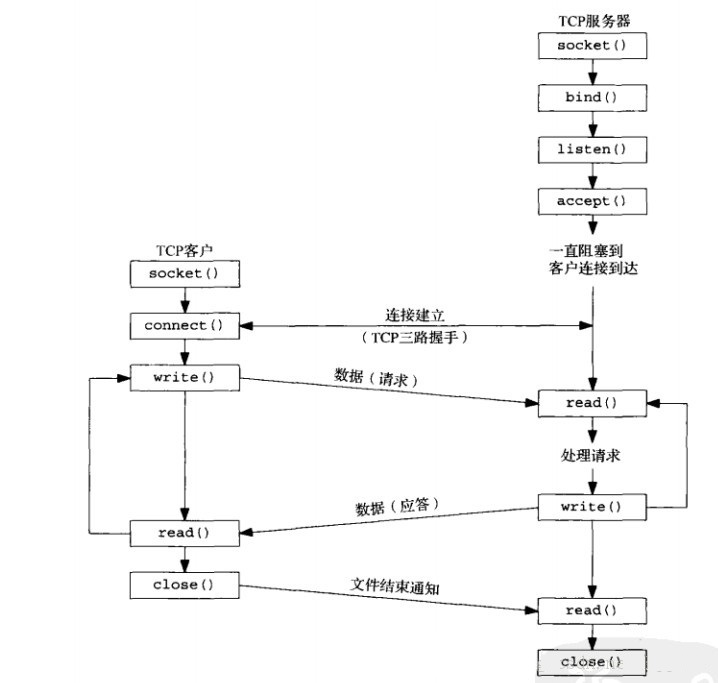
将socket fd设置为nonblock（非阻塞）是在服务器编程中常见的做法，**采用blocking IO并为每一个client创建一个线程的模式开销巨大且可扩展性不佳**（带来大量的切换开销），更为通用的做法是采用线程池+Nonblock I/O+Multiplexing（select/poll，以及Linux上特有的epoll）。

**1. read总是在接收缓冲区有数据时立即返回，而不是等到给定的read buffer填满时返回。**

**只有当receive buffer为空时，blocking模式才会等待**，而nonblock模式下会立即返回-1（errno = EAGAIN或EWOULDBLOCK）

**三. read/write对连接异常的反馈行为：**

对应用程序来说，与另一进程的**TCP**通信其实是**完全异步**的过程

：

**三次握手流程**

**客户端发个请求“开门呐，我要进来”给服务器**

**服务器发个“进来吧，我去给你开门”给客户端**

**客户端有很客气的发个“谢谢，我要进来了”给服务器**

**四次挥手流程**

**客户端发个“时间不早了，我要走了”给服务器，等服务器起身送他**

**服务器听到了，发个“我知道了，那我送你出门吧”给客户端，等客户端走**

**服务器把门关上后，发个“我关门了”给客户端，然后等客户端走（尼玛~矫情啊）**

**客户端发个“我知道了，我走了”，之后自己就走了**

**GateWay网关：网关实质上是一个网络通向其他网络的IP地址。只有设置好网关的IP地址，TCP/IP协议才能实现不同网络之间的相互通信**

**DNS：域名服务器(Domain Name Server)。在Internet上域名与IP地址之间是一一对应的，域名虽然便于人们记忆，但机器之间只能互相认识IP地址**

**VPN英文全称：**Virtual Private Network**（虚拟专用网络）。**

HTTP协议是一种应用层协议，HTTP是HyperText Transfer Protocol(超文本传输协议)的英文缩写。HTTP可以通过传输层的TCP协议在客户端和服务器之间传输数据。

1. **简述七层模型与四层模型的区别？列出常见的网络协议？**  
   七层： OSI 开放系统互联参考模型，它是理论的，参考模型  
   物理层->数据链路层->网络层->传输层->会话层->表示层->应用层  
   四层: TCP/IP模型 ,现在实际应用的这一层  
   网际接口层->互联网层->传输层->应用层

****make是一个命令工具，是一个解释makefile中指令的命令工具****

1、makefile

**makefile**定义了一系列的规则来指定，哪些文件需要先编译，哪些文件需要后编译，哪些文件需要重新编译，甚至于进行更复杂的功能操作，因为 makefile就像一个Shell脚本一样，其中也可以执行操作系统的命令。makefile带来的好处就是--“自动化编译”,一旦写好，只需要一个make命令，整个工程完全自动编译，极大的提高了软件开发的效率。

**GCC**是由GNU之父Stallman所开发的linux下的编译器，全称为GNU Compiler Collection， 目前可以编译的语言包括：C, C++, [Objective-C](https://www.baidu.com/s?wd=Objective-C&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBuj63nvPbuhF9PADkPWnz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHf1PWRsPjbY" \t "https://zhidao.baidu.com/question/_blank), Fortran, Java, and Ada, 可以在其官方页面找到更加详细的信息什么是GCC？GCC是一个原本用于[Unix-like](https://www.baidu.com/s?wd=Unix-like&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBuj63nvPbuhF9PADkPWnz0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHf1PWRsPjbY" \t "https://zhidao.baidu.com/question/_blank)系统下编程的编译器。