



第七章 应用层

课前思考

- 什么是域名系统（DNS）？
- 什么是URL？
- WWW基本要素是什么？
- 电子邮件是怎么工作的？
- 网络文件传输是怎么工作的？
- 应用进程跨越网络的通信是如何进行的？





本章内容

- 7.1 概述
- 7.2 DNS（域名系统）
- 7.3 WWW（万维网）
- 7.4 E-mail（电子邮件）
- 7.5 文件传送协议
- 7.6 远程终端协议
- 7.7 简单网络管理协议
- 7.8 应用进程跨越网络的通信



7.1 概述

- 本章所述应用层是指Internet参考模型的应用层，相当与OSI的会话层、表示层、应用层。
- 应用层处于网络体系结构的最高层，在传输层的支持下，为网络用户提供各种网络服务，包括文件服务，电子邮件，数据库服务，WWW服务，DNS等。
- 应用层意义在于使用户方便快捷地访问网络共享资源，而不必关心实现资源共享的通信细节。
- 应用层主要完成网络数据处理功能。在传输层的支持下，应用层虚拟通信十分简单，通过简单的命令即可实现数据传输。



7.2 DNS（域名系统）

7.2.1 DNS的作用

- Internet上连接无数的计算机，每台计算机都是通过IP地址来标识的。但是**IP地址不便记忆**，用户在访问网络时喜欢用便于记忆的域名来代替**32位IP地址**。例如：
 - 合肥工业大学电子邮件服务器的域名：hfut.edu.cn
 - 合肥工业大学主页的域名：www.hfut.edu.cn
- **网络上的主机和路由器只识别IP地址**，所以需要DNS将域名转化为相应的IP地址。

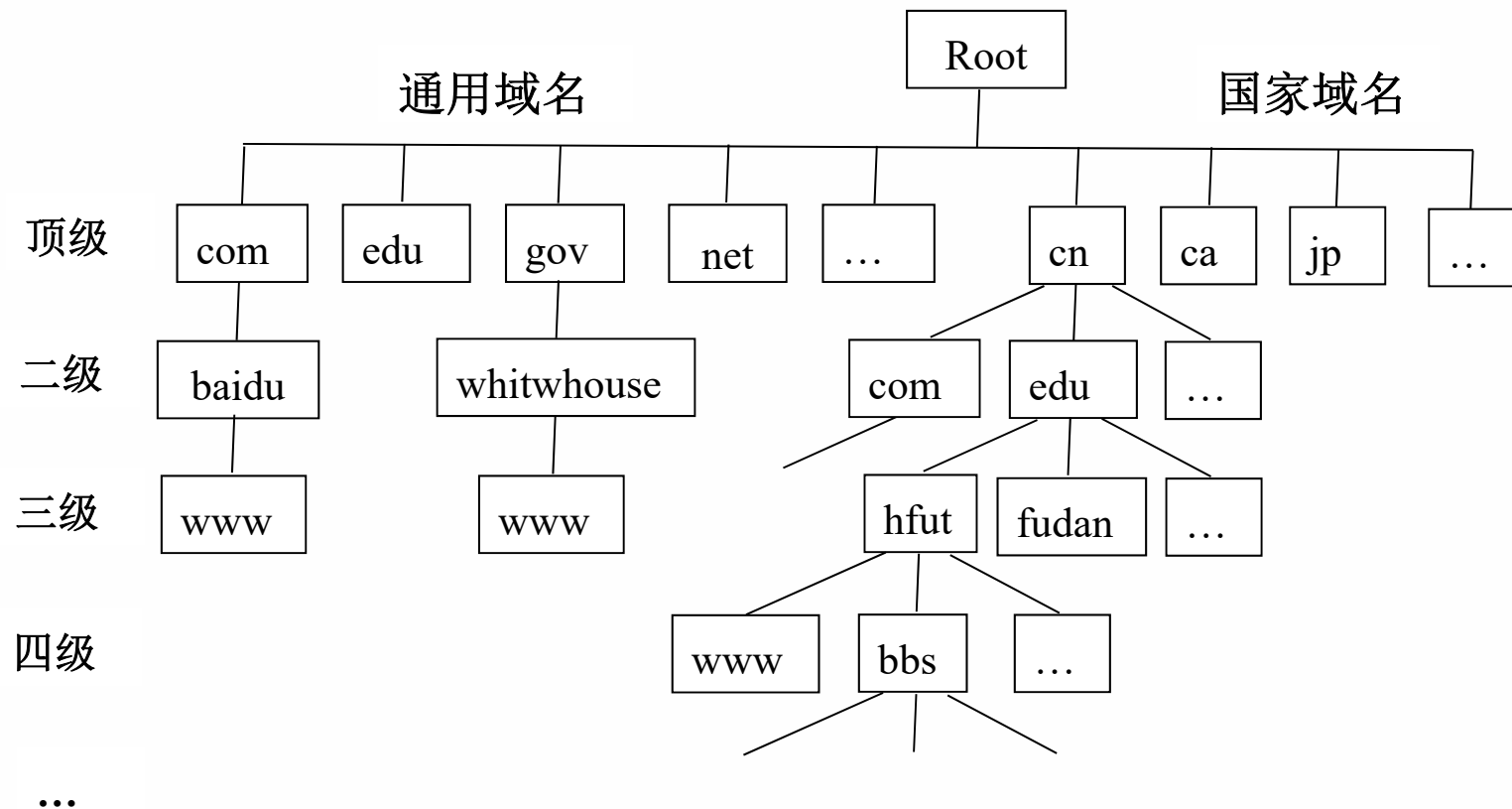


- 一个良好的域名系统应该满足以下条件

- ➔ 首先，主机名称是全局唯一的，即能够在整个因特网通用；
- ➔ 其次，要便于管理，因特网中的主机名称的管理工作包括名称分配、名称确认和名称回收等；
- ➔ 最后，要便于映射，即便于名称与IP地址之间的映射。名字到 IP 地址的解析是由若干个域名服务器程序完成的。域名服务器程序在专设的节点上运行，运行该程序的机器称为域名服务器。



7.2.2 域名体系结构





域名系统结构

7.2.2 域名结构

● 域名格式

....三级域名.二级域名.顶级域名。

例: www.hfut.edu.cn

早期因特网用户很少，只用一级命名就可以标识全部网络用户。此时，采用的域名结构是平面型命名机制。这种命名机制的优点是名称简短，结构简单，每台主机名由一个字符串组成，没有进一步的层次结构。

随着网络用户的大量增加，域名冲突的可能性越来越大，单一的中央管理机构的工作负担也变得越来越重，集中式单一映射表要响应大量主机的映射请求，所以地址映射效率也越来越低，在这种情况下，因特网采用了层次型命名机制。



- 7个传统的通用域名分别是：**.com**（公司企业），**.net**（网络服务机构），**.org**（非盈利组织），**.int**（国际组织），**.edu**（美国教育机构），**.gov**（美国政府部门），**.mil**（美国军事部门）。
- 域名只是一个**逻辑概念**，并不反映计算机所在的**物理位置**，它与**IP子网**也没有关系。例如，计算机系服务器和英语系服务器可能在同一幢大楼内，并共用一个**LAN**，但它们可以有各自的域名。



- **DNS**不区分大小写字母，每级域名不超过**63**个字母或数字，完整的域名不超过**255**个字符。
- **DNS**不限制一个域名包含多少个下级域名，也不规定每一级域名的含义，各级域名由其上一级的域名管理机构管理。例如，要想命名**vlsi.cs.gale.edu**，则需经过**cs**，**gale**，**edu**的批准。



- **Internet**上主机的**域名具有唯一性**，因此**Internet**的域名资源非常有限、宝贵，好的域名具有非常大的商业价值。目前它由全球各组织或政府指定的部门进行管理，并用全球分布的**DNS**服务器进行域名的解析处理。



- 理论上，Internet域名（尤其是国际域名）的注册没有任何限制，只要注册前还没有被注册过就可以。注册可以通过相应的机构或其代理进行，在他们的网站上可以在线注册。如果域名注册中涉及恶意抢注、品牌保护或知识产权等方面纠纷，可以通过世界知识产权组织（WIPO）或我国的工商机构进行仲裁。
- 顶级域名由设在美国的Internet网络信息中心（Internet Network Information Center, InterNIC）负责管理，**顶级域名分为两大类：通用和国家**。一个组织可以挂在国家顶级域，也可挂在通用顶级域名下，但基本上只挂在一个顶级域下。



● 中国互联网络信息中心（CNNIC）是CN域名的注册管理机构，负责运行和管理相应的CN域名系统，维护中央数据库。主要职责如下：

- ➔ （1）运行、维护和管理CN域名服务器和相关资料，保证CN域名系统有效运行。
- ➔ （2）授权CN域名注册服务机构提供CN域名注册服务。作为CN域名注册管理机构，CNNIC不再直接对最终用户提供CN域名注册相关服务，域名注册服务将转由CNNIC认证的域名注册服务机构提供。



- 中国互联网络信息中心（CNNIC）是CN域名的注册管理机构，负责运行和管理相应的CN域名系统，维护中央数据库。主要职责如下：

- ➔ （3）建议为主机确定域名时应尽量使用有意义的字符。
- ➔ （4）一个域名对应一个IP地址，但是一个IP地址可对应多个域名。例如，一台计算机有一个IP地址，但是该主机既可以作为邮件服务器，也可以作为WWW服务器，因而可以有多个域名。
- ➔ （5）主机的IP地址和域名从使用的角度看没有区别。但是，如果使用的系统中没有域名服务器，则只能使用IP地址而不能使用域名。



域名解析

7.2.3 域名解析

- DNS实际上是由许多域名服务器组成的分布式数据库，实现域名与IP地址的解析。
- 域名解析方法
 - 静态解析
 - 动态解析



静态解析

静态解析

- 静态域名解析是通过客户机上的一个解析文件（**hosts**文件）来进行的。
 - ➔ 在 **Windows** 中，该文本文件位于系统文件夹下的 **\system32\drivers\etc** 文件夹中，包含一个域名与**IP**地址的映射表。
- 由于**hosts**文件必须由人工维护，不能及时反映域名的变化，并且管理的工作量也很大。



Host文件

File Explorer window showing the path: 此电脑 > 本地磁盘 (C:) > Windows > System32 > drivers > etc. The file 'hosts' is selected, showing a size of 136 KB.

Notepad++ window showing the contents of the hosts file:

```
1 # Copyright (c) 2014-2016, racaljk.
2 # https://github.com/racaljk/hosts
3 # Last updated: 2016-11-13
4
5 # This work is licensed under a CC BY-NC-SA 4.0 International License.
6 # https://creativecommons.org/licenses/by-nc-sa/4.0/
7
8 # Localhost (DO NOT REMOVE)
9 127.0.0.1    localhost
10 ::1 localhost ip6-localhost ip6-loopback
11
12 # Modified hosts start
13
14 # Armorgames Start
15 93.184.220.39  cache.armorgames.com
16 93.184.220.39  gamemedia.armorgames.com
17 93.184.220.39  quests.armorgames.com
18 93.184.220.39  armatars.armorgames.com
19 93.184.220.39  agi.armorgames.com
```

Notepad++ status bar: 5 Normal text file length: 139024 lines Ln: 1 Col: 1 Sel: 0 | 0 Dos/Windows UTF-8



动态域名解析

动态域名解析

- DNS实际上是由许多域名服务器组成的分布式数据库，实现域名与IP地址的解析。
- 域名解析过程
 - 首先由本地域名服务器解析，如果本地没有找到，则递交给根域名服务器解析，根域名服务器根据待解析的域名，找到下属的授权域名服务器，并将授权域名服务器的IP地址返回给本地域名服务器。
 - 本地域名服务器直接向授权域名服务器进行查询，得到域名对应的IP地址。
 - 本地域名服务器维护一个高速缓存，存放最近用过的域名以及从何处得到其IP地址的记录。这样，可通过高速缓存直接找到根域名服务器下属的授权域名服务器，从而减轻根域名服务器的负担。



域名缓存机制的益处

- (1) 域名解析的高效性：本地的一个客户端在解析过某个域名后，其他的客户端对该域名的解析就可以在本地的域名解析服务器缓存中快速命中，极大提高了域名解析的效率。
- (2) 域名解析处理的分布性：通过地理上分散的域名解析服务器缓存的解析处理，将最终的域名解析处理分布到当地（或就近）的域名解析服务器进行处理，极大缓解了该域名管理服务服务器的解析处理压力。
- (3) 域名解析的可靠性：通过分布的解析服务器的协同处理，提高了整个域名解析系统的可靠性，局部的域名服务器故障不会导致整个域名服务系统的瘫痪。



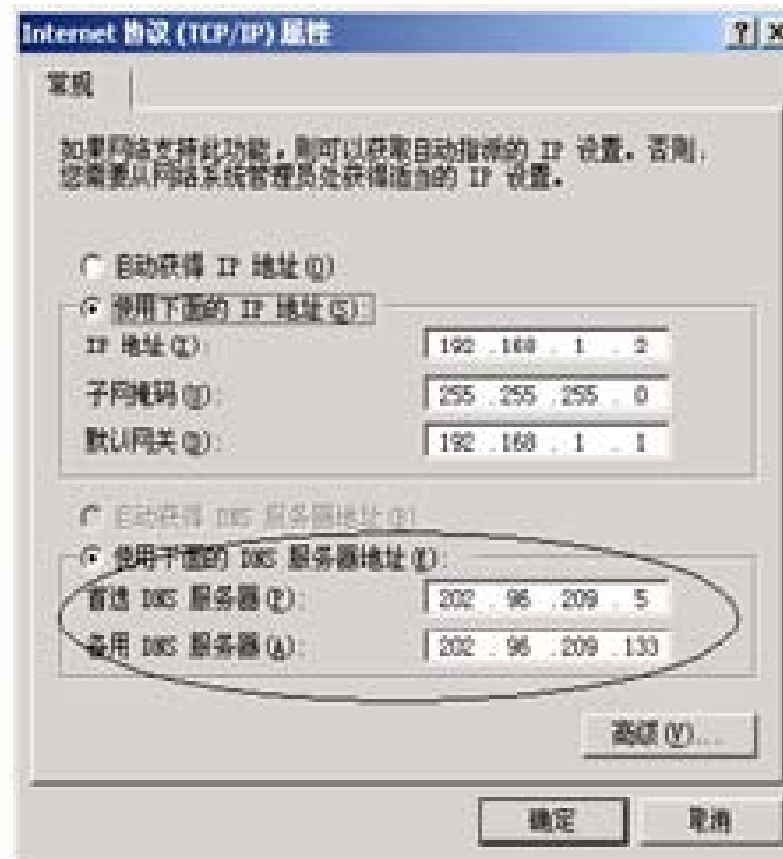
域名的管理和解析总结

- 要实现DNS域名解析，网络系统中必须至少设置一台服务器作为**DNS的解析服务器**，同时让客户端的DNS域名解析都指向这台服务器；
- 要实现本域及下级子域的管理，网络系统中必须至少设置一台服务器作为**DNS的管理服务器**，同时让上级域名管理服务器有关这个域的管理都赋予这台服务器。
- 实际应用中，这两种服务器通常都**合二为一**，笼统地称为**DNS服务器**。



域名服务器

● 域名服务器



指定要使用的DNS服务器



7.3 WWW(万维网)

7.3.1 概述

- WWW(World Wide Web)即全球信息网，简称Web，又称为万维网。
- WWW由Internet上所有Web服务器提供的网页组成，不同的网页之间通过超链接连接，客户端使用一个标准的浏览器就能找到想要的网页，再点击网页中的超链，链接到另一网页；这一过程可以无限重复，直到找到所需的所有网页。



- 1989年WWW起源于CERN(欧洲粒子研究中心)，最初是为了参与核物理实验而分布在不同国家的科学家之间交流研究报告、装置蓝图、图画、照片和其他文档而设计的一种网络通信工具。
- 1989年3月，物理学家Tim Berners—Lee提出初步的研究报告，18个月后有初始的系统原型。
- 1993年2月发布了第一个图形式的浏览器Mosaic，它的作者 Marc Andresen 在 NCSA （ National Center for Supercomputing Applications）成立了一个公司，这就是首开 Web 服务器先河的网景通信公司（ Netscape Communications）。

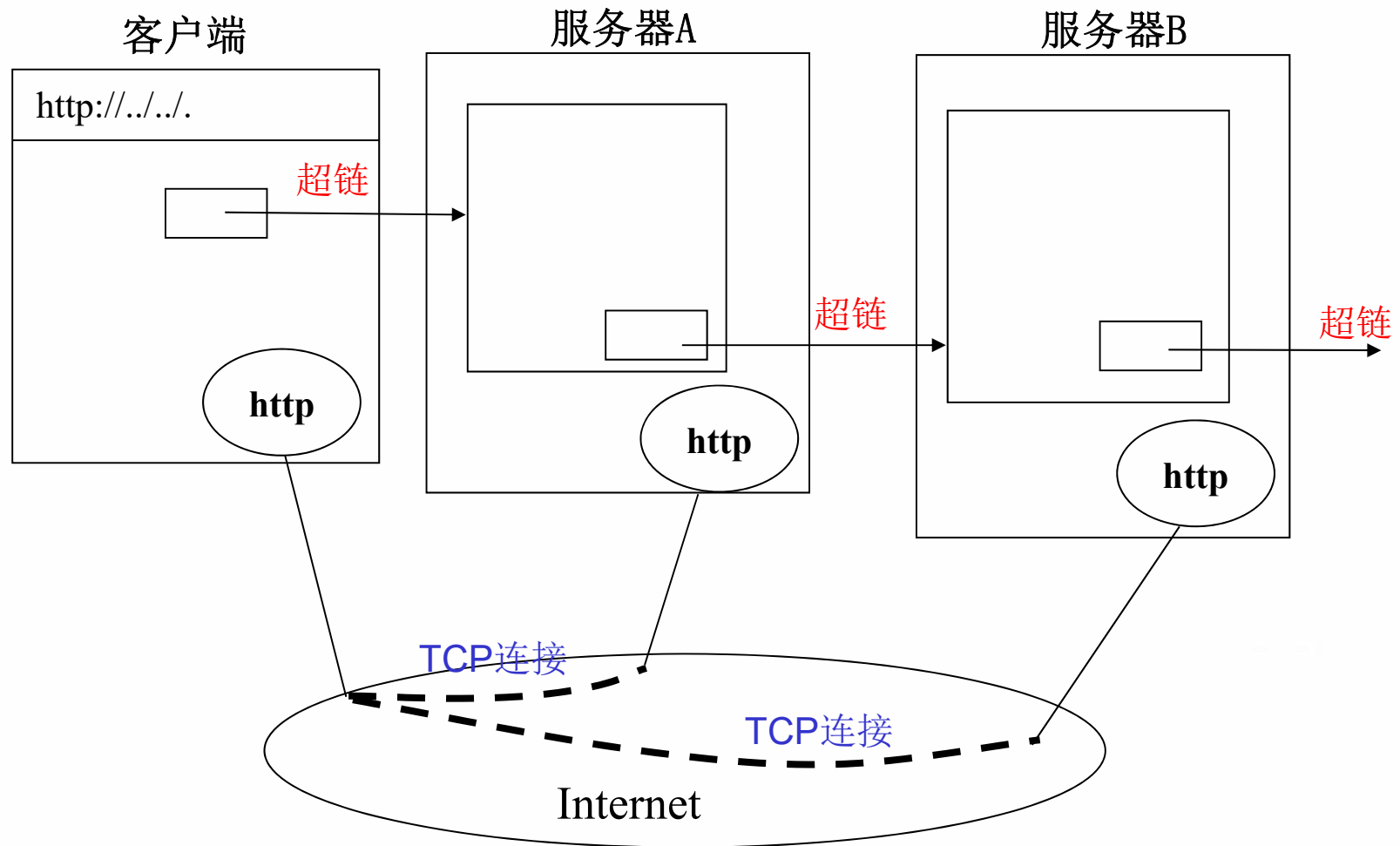


Web技术

- Web技术是一种综合性网络应用技术，关系到网络信息的表示、组织、定位、传输、显示以及客户和服务器的交互作用等。
- 通常文字信息组织成线性的ASCII文本文件，而Web上的信息组织成非线性的超文本文件（Hypertext）。简单地说，超文本可以通过超链接（Hyperlink）指向网络上的其他信息资源。
- 超文本互相链接成网状结构，使得人们可以通过链接追索到与当前节点相关的信息。这种信息浏览方法正是人们习惯的联想式、跳跃式的思维方式的反映。
- 一个超文本文件叫做一个网页（WebPage），网页中包含指向有关网页的指针（超链接）。如果用户选择了某一个指针，则有关的网页就显示出来。超链接指向的网页可能在本机，也可能在网上别的地方。



WWW服务模型





- Web上的信息不仅是超文本文件，还可以是语音、图形、图像、动画等。就像通常的多媒体信息一样，这里有一个对应的名称叫超媒体（Hypermedia）。
- 超媒体包括了超文本，也可以用超链接连结起来，形成超媒体文档。
- 超媒体文档的显示、搜索、传输功能全都由浏览器（browser）实现。



- WWW本质上是一种客户机/服务器（C/S）技术，其体系结构提供了一个灵活且强有力的设计模型：应用
软件采用标准数据格式显示，并能通过它的客户机-浏
览器（**Browser**）进行浏览。
- 浏览器是一个网络应用软件，它能向统一命名的数据
服务对象的网络服务器——WWW服务器（或Web服
务器）发送请求。而WWW服务器采用标准格式编码
的响应来回复请求，WWW服务模型有时也称为浏览
器/服务器（**B/S**）模型。



- 常用的WWW服务器有Microsoft IIS等
- 万维网必须解决以下几个问题
 - ➔ (1) 怎样标志分布在整个因特网上的万维网文档?
 - ➔ (2) 用何协议实现万维网上各种超链的链接?
 - ➔ (3) 怎样使各种万维网文档都能在因特网上的各种计算机上显示出来, 同时使用户清楚地知道在什么地方存在着超链?
 - ➔ (4) 怎样使用户能够很方便地找到所需的信息?

URL (统一资源定位符)



URL

URL (统一资源定位符)

- URL 俗称“**网址**”，用于定位Internet上可访问的资源。
- 是WWW的一种混合语，它表示要访问的主机地址、获取服务所用的协议以及所要浏览文件的路径和名字。
- **URL** 相当于一个文件名在网络范围的扩展。因此 **URL** 是与因特网相连的机器上的任何可访问对象的一个指针。



URL格式

URL (统一资源定位符)

● URL格式

<协议>://<域名>/<路径>

例: <http://ci.hfut.edu.cn/main.htm>

<http://ci.hfut.edu.cn/fjs/list.htm>

- 第一部分是协议（或称为服务方式）；
- 第二部分是存有该资源的主机IP地址（有时也包括端口号）；
- 第三部分是主机资源的具体地址，如路径等。
- 第一部分和第二部分之间用“://”符号隔开，冒号和两个斜线是规定的格式；第二部分和第三部分用“/”符号隔开。第一部分和第二部分是不可缺少的，第三部分有时可以省略。
- 主机是存放资源的主机在因特网中的域名，端口路径有时可省略，若再省略文件的<路径>项，则 URL 就指到因特网上的某个主页（home page）或IP地址。



URL访问过程

URL (统一资源定位符)

如果用户选择了一个要访问的网页，则浏览器和Web服务器的交互过程如下：

- (1) 浏览器计算出URL。
- (2) 浏览器通过DNS服务器查找URL的IP地址。
- (3) DNS给出IP地址。
- (4) 浏览器与主机的端口80建立TCP连接。
- (5) 浏览器发出请求GET/main.html文件。
- (6) ci.hfut.edu.cn服务器发送main.html文件。
- (7) 释放TCP连接。
- (8) 浏览器显示main.html文件。



URL (统一资源定位符)

- 其中第(5)步的“GET”是HTTP协议提供的少数操作方法中的一种，其含义是读一个网页。
- 常用的还有HEAD（读网页的头信息）和POST（把消息加到指定的网页上）等。
- 另外，要说明的是很多浏览器不但支持HTTP协议，还支持FTP、Telnet、Gopher等，使用方法与HTTP完全一样。正因为如此，很多人认为WWW是无所不能的。



URL (统一资源定位符)

URL最常用的访问方式为以下三种。

- (1) **ftp**: 文件传送协议 **FTP**。
- (2) **http**: 超文本传送协议 **HTTP**。
- (3) **News**: **USENET** 新闻。



URL (统一资源定位符)

对于万维网的网点的访问要使用HTTP协议，**HTTP**的默认端口号是**80**，通常可省略。若再省略文件的<路径>项，则 **URL** 就指到因特网上的某个主页（**home page**）。主页可以是以下几种情况之一：

- (1) 一个**WWW**服务器的最高级别的页面。
- (2) 某一个组织或部门的一个定制的页面或目录，从这样的页面可链接到因特网上的与本组织或部门有关的其他站点。
- (3) 由某一个人自己设计的描述他本人情况的**WWW**页面。

例如，要查到有关合肥工业大学的信息，应先进入合肥工业大学的主页 **http://www.hfut.edu.cn**，这里省略了默认的端口号。而从主页入手，就可以链接到因特网上有关合肥工业大学各个部门消息的页面。



URL (统一资源定位符)

- URL最大的缺点是当信息资源的存放地点发生变化时，必须对URL作相应的改变。
- 因此人们正在研究新的信息资源表示方法。例如，URI (Universal Resource Identifier) 即“通用资源标识” (参见RFC 1630)、URN (Uniform Resource Name) 即“统一资源名”和 URC (Uniform Resource Citation) 即“统一资源引用符”等。



超文本传输协议

超文本传输协议

- 能够快速查找这些超文本文档的协议，即 **HTTP 协议**（**Hypertext Transfer Protocol**）。
- HTTP 是一个应用层协议，是面向事务（**transaction-oriented**）的应用层协议，它是万维网上能够可靠交换文件（包括文本、声音、图像等各种多媒体文件）的重要基础。



HTTP：超文本传输协议。

- 使用 TCP 连接进行可靠的传送。

该协议基于TCP和C/S方式，服务器端口号为80；客户端只要与服务器的80端口建立一个TCP连接，就能够通过HTTP报文进行会话，实现超文本传输。



超文本传输协议

- HTTP的主要特点

- ➔ HTTP 是面向事务的客户服务器协议
- ➔ 万维网浏览器就是一个 HTTP 客户，而在万维网服务器等待 HTTP 请求的进程常称为 **HTTP daemon**，有的文献将它缩写为 **HTTPD**；**HTTP daemon** 在收到 **HTTP** 客户的请求后，把所需的文件返回给 **HTTP** 客户。



HTTP操作过程

超文本传输协议

- 一次HTTP操作称为一个事务，其工作过程可分为四步：
 - ➔ (1) 首先客户机与服务器需要**建立连接**。只要单击某个超级链接，客户进程建立一条同服务器进程的 **TCP 连接**。
 - ➔ (2) 建立连接后，**客户**机发送一个**请求**给服务器，请求方式的格式为：统一资源标识符（**URL**）、协议版本号等。



超文本传输协议

- 一次HTTP操作称为一个事务，其工作过程可分为四步：
 - (3) 服务器接到请求后，给予相应的响应信息，其格式为一个状态行，包括信息的协议版本号、一个成功或错误的代码，后边是MIME信息包括服务器信息、实体信息和可能的内容。
 - (3) 客户端接收服务器所返回的信息，通过浏览器显示在用户的显示屏上，然后客户机与服务器断开连接。



超文本传输协议

- 如果在以上过程中的某一步出现错误，那么产生错误的信息将返回到客户端，由显示屏输出。
- 对于用户来说，这些过程是由**HTTP**自己完成的，用户只要用鼠标单击，等待信息显示即可。



HTTP报文格式

● HTTP报文格式

HTTP报文包括请求报文和响应报文。

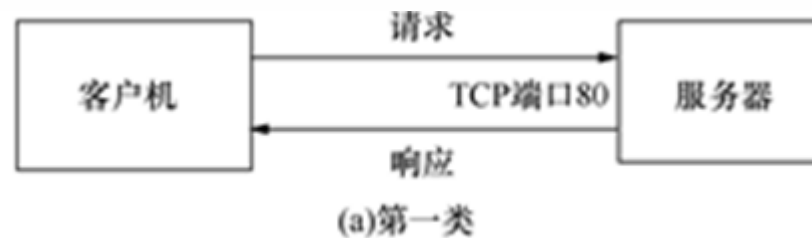
HTTP报文面向字符，每个字段都是一些ASCII字符串。



HTTP中的请求-响应链

HTTP中有三类请求-响应链

- 第一类：客户机直接向服务器发送请求，建立TCP连接，默认端口是80，然后在这个连接上发送请求。服务器监听连接上的请求，对于每一个请求，服务器启动一个线程或进程来对它进行服务。一旦请求处理完毕，服务器把响应在这个连接上送回给原来的客户机。





HTTP中有三类请求-响应链

- 第二类：通过代理服务器（**Proxy Server**），用户向代理服务器建立连接并发送请求，代理服务器再向WWW服务器发送请求，WWW服务器向代理服务器回送响应，最后代理向用户发回响应。



(b)第二类



HTTP中有三类请求-响应链

- 第三类：隧道，和第二类相似，也有一个中间节点。隧道和代理不同，它只是一个用户向WWW服务器发送请求以及从服务器接收响应的通道，不执行其他任何功能（如代理的cache功能、用户鉴别功能等）。代理和隧道可以是多重的，即在客户机到WWW服务器之间可以有多个代理和隧道。



超文本传输协议

- 客户请求是简单的 ASCII 文本，服务器的响应也是以 ASCII 文本开始（首部）的，后面跟着数据（可以是 ASCII 或二进制数据）。
- 客户程序软件（浏览器）分析服务器的响应，并把它格式化输出，同时以高亮方式显示指向其他文档的链接。



超文本传输协议

- HTTP带来的最大的性能问题是每个文件使用一条 TCP 连接。当客户请求报文的长度超过服务器通告的MSS时，缓慢的建立使每一个TCP连接增加了额外的时延。
- 另一个问题是服务器进程正常关闭连接将引起在服务器主机上产生TIME-WAIT时延，在一个繁忙的服务器上可以看到很多这种待终止的连接。



超文本标记语言

万维网的文档

- 超文本标记语言（**HyperText Markup Language, HTML**）使得万维网页面的设计者可以很方便地用一个超链从本页面的某处链接到因特网上的任何一个万维网页面。



万维网的文档

- 仅当 HTML文档是以.html 或 .htm为后缀时，浏览器才对此文档的各种标签进行解释。
- 如 HTML文档改换以.txt 为其后缀，则 HTML 解释程序就不对标签进行解释，而浏览器只能看见原来的文本文件。
- 当浏览器从服务器读取 HTML 文档后，就按照 HTML 文档中的各种标签，根据浏览器所使用的显示器的尺寸和分辨率大小，重新进行排版并恢复出所读取的页面。



万维网的文档

- **HTML**（超文本标注语言）用于编写网页；其特点是：

- 定义了多种排版命令
- 允许插入图象
- 支持超级链接

- 页面

页面是由**HTML**（超文本标注语言）编写的超文本文件，它能够集成文本、图象、语音和超链为一体。



超文本标记语言

万维网的文档

- 超文本标记语言HTML中的Markup的意思就是“设置标记”。HTML 定义了许多用于排版的命令（标签）。HTML把各种标签嵌入到万维网的页面中，这样就构成了所谓的 HTML文档，它是一种可以用任何文本编辑器创建的 ASCII码文件。



万维网的文档

- 元素（**element**）是HTML文档结构的基本组成部分。一个HTML文档本身就是一个元素。
- 每个HTML文档由两个主要元素组成：首部（**head**）和主体（**body**）
 - ➔ 首部包含文档的标题（**title**），以及系统用来标识文档的一些其他信息。标题相当于文件名。
 - ➔ 主体是HTML文档最主要的部分。主体部分往往又由若干个更小的元素组成，如段落（**paragraph**）、表格（**table**）和列表（**list**）等。



万维网的文档

HTML用一对标签（即一个开始标签和一个结束标签）或几对标签来标识一个元素。开始标签由一个小于字符“<”、一个标签名和一个大于字符“>”组成。结束标签和开始标签的区别只是在小于字符的后面要加上一个斜线字符“/”。标签名并不区分大写和小写。

下面是一个简单的例子，用来说明HTML文档中标签的用法。

```
<HTML>           { 文档开始 }
<HEAD>           { 首部开始 }
<TITLE>一个HTML的例子</TITLE> { “一个HTML的例子”是文档的标题 }
</HEAD>          { 首部结束 }
<BODY>           { 主体开始 }
  <H1>HTML 很容易掌握</H1> { 1 级标题 }
  <P>这是第一个段落。</P>   { <P>和</P>之间的文字是一个段落 }
  <P>这是第二个段落。</P>   { <P>和</P>之间的文字是一个段落 }
</BODY>          { 主体结束 }
</HTML>          { 文档结束 }
```



● 请求报文：从客户端向服务器发送请求报文

➤ 请求报文格式

方法	S	URL		S	HTTP版本	CR	请求行
头部字段名	:	S	值	CR		头部行	
.....							
头部字段名	:	S	值	CR			
CR							
实体主体(通常不用)							

• 请求行

方法：即请求的操作命令，包括：

GET：请求发送URL指定的页面。如
`GET/hypertext/www/project.html`

HEAD：请求服务器仅发送一个页面头部信息（如修改时间、大小等）。

PUT：向URL指定的位置上传一个页面，用于远程创建网页。



● 请求报文: 从客户端向服务器发送请求报文

➤ 请求报文格式

方法	S	URL		S	HTTP版本	CR	请求行
头部字段名		:	S	值		CR	
.....							
头部字段名		:	S	值		CR	
CR							
实体主体(通常不用)							

• 请求行

方法: 即请求的操作命令, 包括:

POST: 向网页中添加数据。

DELETE: 删除网页。

TRACE: 用于回路测试 (服务器将请求报文返回)。

CONNECT: 用于代理服务器。



URL: 统一资源定位符

版本: 如HTTP/1.0, HTTP/1.1

CR: 回车; S: 空格

方法	S	URL		S	HTTP版本	CR	请求行
头部字段名		:	S	值		CR	
.....							
头部字段名		:	S	值		CR	
CR							
实体主体(通常不用)							



头部行：用于说明浏览器、服务器或报文主体的一些信息。

例如：

Host: www.hfut.edu.cn {给出主机名}

Connection: close {响应该请求后，释放连接，即非持续}

User-Agent: Mozilla/5.0 {使用Netscap浏览器}

Accept-language: cn {用户希望优先得到中文版本文档}

头部字段名	:	S	值	CR	} 头部行
.....					
头部字段名	:	S	值	CR	



- 状态行

状态码表示对请求的响应状态；原因短语是对状态码的解释。

类	状态码	原因短语
成功	200	OK（成功）
	201	Created（生成）
	202	Accepted（已接受）
	204	No Content（无内容）
重定向	301	Moved Permanently（已永久转移）
	302	Moved Temporarily（已临时转移）
	304	Not Modified（没有修改）
客户端差错	400	Bad Request（不正确请求）
	401	Unauthorized（未授权）
	403	Forbidden（被禁止）
	404	Not Found（没有找到）
服务器差错	500	Internal Server Error（内部服务器错误）
	501	Not Implemented（未实现）
	502	Bad Gateway（不正确网关）
	503	Service Unavailable（服务不可用）



- 状态行：用于表达附加信息，如服务器位置、服务器开发者等。
如：

HTTP/1.1 301 Moved Permanently {状态行，指示重定向响应报文}
Location: <http://www.gs.fudan.edu.cn/index.htm> {给出服务器的位置}

- 实体主体

即响应返回的内容，如一个HTML文档。

方法	S	URL		S	HTTP版本	CR	请求行
头部字段名	:	S	值	CR		} 头部行	
.....							
头部字段名	:	S	值	CR			
CR							
实体主体(通常不用)							



- 动态网页

服务器应用程序根据浏览器请求，动态生成HTML文档。

- 活动网页

当浏览器请求时，服务器返回一段程序在浏览器上运行，更新网页内容。



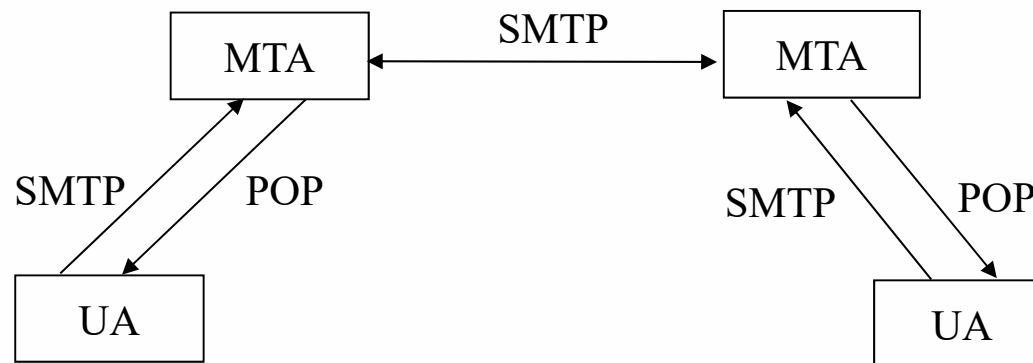
7.4 E-mail (电子邮件)

- 电子邮件 (E-mail) 是目前Internet上使用最频繁的服务之一。
- 电子邮件是以电子方式发送传递的邮件。只要通信双方都有电子邮件地址，便可以电子传播为媒介，交互邮件。
- Internet上电子邮件系统采用客户机/服务器模式，信件的传输通过相应的软件来实现，这些软件要遵循有关的邮件传输协议。



7.4 E-mail（电子邮件）

7.4.1 电子邮件系统的基本组成



- UA（User Agent，用户代理）

该客户端软件用来编写、显示、与本地邮件服务器的通信（发送/接受邮件）。如：Outlook Express, Foxmail等。

- MTA（Message Transfer Agent，消息传输代理）

该服务器软件用来传递信件，向发件人报告邮件传送情况（已交付、被拒绝、丢失等）。如：Microsoft exchange Server, Sandmail等。



7.4 E-mail（电子邮件）

- 传送电子邮件时使用的协议
 - ➔ 简单邮件传输协议（**Simple Mail Transport Protocol, SMTP**）：用于电子邮件发送服务，也即用于MTA之间以及UA向MTA传输邮件。
 - ➔ 邮局协议（**Post Office Protocol, POP**），用于电子邮件接收服务，也即用于收件人从本地邮件服务器中读取邮件。
 - ➔ 还有其他的通信协议，在功能上它们与上述协议是相同的。



● 电子邮件地址

- 用户在Internet上收发电子邮件时，必须拥有一个电子信箱（Mailbox），而每个电子信箱有一个唯一的地址，通常称为电子邮件地址（E-mail Addresses）。
- E-mail地址由两部分组成，以符号“@”分隔，“@”前面的部分是用户名，“@”后面的部分为邮件服务器的域名，如E-Mail地址“yuqi.fan@qq.com”中，“yuqi.fan”是用户名，“qq.com”为网易的邮件服务器的域名。

● 电子邮件工具

- 电子邮件应用程序很多，常见的有Foxmail、Outlook等。



SMTP命令和传递邮件过程

7.4.2 SMTP命令和传递邮件过程

- SMTP是目前使用最广泛的邮件传输协议，该协议基于TCP和C/S方式，服务器端口号为25。
- SMTP发送E-mail的基本思想是接收端（即服务器）在TCP的25号端口等待发送端的信息，发送端向接收端发出连接请求，一旦连接成功，即进行邮件信息的交换，邮件传递结束后释放连接。



7.4.2 SMTP命令和传递邮件过程

- 由于SMTP使用客户机/服务器方式，因此负责发送邮件的SMTP进程就是SMTP客户，而负责接收邮件的SMTP进程就是SMTP服务器。



7.4.2 SMTP命令和传递邮件过程

- SMTP提供了这样一种邮件传输机制
 - 当收件方和发件方都在同一个网络上时，可以将邮件直接传送给对方；
 - 当对方不在同一个网络上时，需要通过一个或者几个中间服务器来进行转发SMTP。



7.4.2 SMTP命令和传递邮件过程

- 首先由发件方提出申请，要求与接收方SMTP建立双向通信渠道，收件方可以是最终收件人，也可以是中间转发服务器
- 收件方服务器确认可以建立连接以后，双方就可以开始通信了，发件方SMTP向收件方SMTP发出MAIL命令，告知发件方的身份，如果收件方接收，就回答OK。



7.4.2 SMTP命令和传递邮件过程

- 发件方再发出**RCPT**命令，告知收件人的身份，收件方SMTP确认是否接收或转发，如果同意，就回答**OK**，接下来就可以进行数据传输了。
- 在通信过程中，**发件方SMTP与收件方SMTP采用对话式交互方式**，即发件方提出要求，收件方进行确认，确认以后才进行下一步的动作，整个过程由发件方控制，有时需要确认几回才能够实现。



7.4.2 SMTP命令和传递邮件过程

● SMTP命令

SMTP面向字符，包括14条命令和21种应答组成，常用命令如下：

SMTP命令	命令格式	命令含义
HELP	HELP <回车>	要求接收方给出相关帮助信息
HELO	HELO <发送方域名> <回车>	告诉接收方自己的E_mail域名
MAIL FROM:	MAIL FROM: <发送方E_mail地址> <回车>	将发送方E_mail地址传送给对方
RCPT TO:	RCPT TO: <接收方E_mail地址> <回车>	将接收方E_mail地址传送给对方;若有多个接收方,可多次使用本命令
DATA	DATA <回车> <回车> . <回车>	用于传输邮件内容。用第一列为“.”且其后为<回车>表示结束。
QUIT	QUIT <回车>	结束邮件传输，释放邮件连接。



● SMTP传递邮件过程

例: **S: Helo hfut.edu.cn**

R: 250 hfut.edu.cn Hello hfut.edu.cn , please to meet you

S: MAIL FROM: <yuqi.fan@hfut.edu.cn>

R: 250 yuqi.fan@hfut.edu.cn ... Sender ok

S: RCPT TO: <wang@pku.edu.cn>

R: 250 wang@pku.edu.cn ... Recipient ok

S: RCPT TO: <zhang@tsinghua.edu.cn>

R: 550 zhang@tsinghua.edu.cn ... No such recipient

S: DATA

R: 354 Enter mail,end with "." on a line by itself

S: Blah Blah Blah

S:

S: .

R: 250 ok

S: QUIT

R: 221 hfut.edu.cn closing connection

应答代码**250**表示请求正常并已完成; **354**提示输入邮件内容; **550**表示找不到邮箱。其他代码请查阅有关RFC文档。

70



电子邮件的信息格式

7.4.2 SMTP命令和传递邮件过程

电子邮件的信息格式

- 一个电子邮件分为信封和内容两大部分。
- **RFC 822**只规定了邮件内容中的首部（**header**）格式，而对邮件的主体（**body**）部分则让用户自由撰写。



7.4.2 SMTP命令和传递邮件过程

电子邮件的信息格式

- 邮件内容首部包括一些关键字，后面加上冒号。最重要的关键字是：**To**和**Subject**。
 - “**To:**”后面填入一个或多个收信人的电子邮件地址。用户只需打开地址簿，点击收信人名字，收信人的电子邮件地址就会自动填入到合适的位置上。
 - “**Subject:**”是邮件的主题。它反映了邮件的主要内容，便于用户查找邮件。



7.4.2 SMTP命令和传递邮件过程

电子邮件的信息格式

- 邮件内容首部包括一些关键字，后面加上冒号。最重要的关键字是：**To**和**Subject**。
 - “抄送： ” 表示应给某某人发送一个邮件副本。
 - “**From:** ” 和 “**Date:** ” 表示发信人的电子邮件地址和发信日期。
 - “**Reply-To:** ” 是对方回信所用的地址。



POP3命令和读取邮件过程

7.4.3 POP3命令和读取邮件过程

- POP主要用于电子邮件的接收，使用**TCP**的**110**端口。
POP也使用**客户机/服务器的工作方式**。在接收邮件的用户 PC 机中必须运行 POP 客户程序，而在用户所连接的 ISP 邮件服务器中则运行 POP 服务器程序。
- POP是一个非常简单、但功能有限的邮件读取协议，现在使用的是它的第三个版本 POP3（ RFC1939 ）。



7.4.3 POP3命令和读取邮件过程

- 当客户机需要服务时，客户端的软件（比如**Outlook**或**Foxmail**），将与**POP3**服务器建立**TCP**连接。
- 此后，需要经过**POP3**协议的三种工作状态：
 - **认证过程**，确认客户机提供的用户名和密码；
 - **处理状态**，用户在该状态下可以收取自己的邮件或者删除邮件；
 - 客户机发出**QUIT**命令，此后进入**更新状态**，将做了删除标记的邮件从服务器删除。



7.4.3 POP3命令和读取邮件过程

● POP3命令

POP3常用命令如下：

POP3命令	命令格式	命令含义
USER	USER <userid> <回车>	给出用户标识
PASS	PASS <password> <回车>	给出用户口令
LIST	LIST [<邮件编号>] <回车>	给出指定或全部邮件的头部信息
DELE	DELE <邮件编号> <回车>	删除指定邮件
RETR	RETR <邮件编号> <回车>	将指定邮件从服务器传到客户机
QUIT	QUIT <回车>	释放POP3连接。



IMAP

7.4.4 IMAP

- 交互式邮件存取协议（Internet Message Access Protocol, **IMAP**）也是按**客户机/服务器方式工作**，现在较新的版本是**IMAP4**。
- 用户在自己的主机上就可以操纵 **ISP** 邮件服务器的邮箱，就像在本地操纵一样。因此 **IMAP** 是一个**联机协议**。当用户的 **IMAP** 客户程序打开 **IMAP** 服务器的邮箱时，就可看到邮件的首部。若需要打开某个邮件，则该邮件才传到用户的计算机上。



7.4.4 IMAP

- **IMAP**最大的好处就是用户可以在不同的地方使用不同的计算机随时上网阅读和处理自己的邮件。
- **IMAP** 还允许收信人只读取邮件中的某一个部分。例如，收到了一个带有视像附件（此文件可能很大）的邮件。为了节省时间，可以先下载邮件的正文部分，待以后有时间再读取或下载这个很长的附件。



7.4.4 IMAP

- **IMAP** 的缺点是如果用户没有将邮件复制到自己的 **PC** 机上，则邮件一直存放在 **IMAP** 服务器上。因此用户需要经常与 **IMAP** 服务器建立连接。



7.4.4 IMAP

● IMAP4是IMAP的最新版本，其主要特征包括：

- (1) 与因特网消息标准完全兼容；
- (2) 许多计算机可以同时访问和管理邮件；
- (3) 允许通过低效率的文件访问协议来进行访问；
- (4) 提供对在线、离线以及无连接三种访问方式的支持；
- (5) 支持共享邮箱的并发访问；
- (6) 客户端软件不需要了解服务器上的文件存储格式。



7.4.4 IMAP

- 必须注意的是不要将邮件读取协议POP或IMAP与邮件传送协议 SMTP 弄混。
 - ⑩ 发信人的用户代理向源邮件服务器发送邮件，以及源邮件服务器向目的邮件服务器发送邮件，都是使用 SMTP 协议。
 - ⑩ POP协议或IMAP协议则是用户从目的邮件服务器上读取邮件所使用的协议。



通用因特网邮件扩充MIME

7.4.5 通用因特网邮件扩充MIME

SMTP有以下缺点：

- (1) SMTP不能传送可执行文件或其他的二进制对象。
- (2) SMTP限于传送7位的ASCII码。许多其他非英语国家的文字（如中文、俄文，甚至带重音符号的法文或德文）就无法传送。
- (3) SMTP 服务器会拒绝超过一定长度的邮件。
- (4) 某些SMTP的实现并没有完全按照RFC 821的SMTP标准。



7.4.5 通用因特网邮件扩充**MIME**

- **MIME**没有改动**SMTP**或取代**SMTP**。
- **MIME**继续使用目前的**RFC 822**格式，但增加了邮件主体的结构，并定义了传送非**ASCII** 码的编码规则。
- **MIME**协议说明了如何安排消息格式使消息可以在不同邮件系统内进行交换，**MIME**格式灵活，它允许邮件中包含任意类型的文件，可以包含文本、图像、声音、视频以及其他应用程序的特定数据。



7.4 E-mail（电子邮件）

7.4.5 通用因特网邮件扩充**MIME**

- 三种常用的内容传送编码（**Content-Transfer-Encoding**）

（1）最简单的编码就是**7位ASCII码**，而每行不能超过**1000**个字符。**MIME**对这种由**ASCII**码构成的邮件主体不进行任何转换。

（2）另一种编码称为**quoted-printable**，这种编码方法适用于当所传送的数据中只有少量的非**ASCII**码的情况。

（3）对于任意的二进制文件，可用**base64**编码。



7.4.5 通用因特网邮件扩充**MIME**

- **MIME**的标准规定**Content-Type**说明必须含有两个标识符，即内容类型（**type**）和子类型（**subtype**），中间用“/”分开。
- **MIME** 标准定义了7种基本内容类型和 15 种子类型。
- 除了内容类型和子类型，还允许发信人和收信人自己定义专用的内容类型。



7.5 文件传送协议

- 文件传输协议（File Transfer Protocol, FTP）通过FTP程序（服务器程序和客户端程序）在Internet上实现远程文件的传输，是因特网上使用得最广泛的文件传送协议。
- FTP实际上就是将各种类型的文件都放在FTP服务器中，用户计算机上要安装一个客户端FTP服务程序，通过这个程序实现对FTP服务器的访问。
- 当通过FTP客户端程序登录FTP服务器时，要求正确回答用户名和口令，才能取得访问权。FTP的主要功能是减少或消除在不同操作系统下处理文件的不兼容性。



- **FTP**是Internet上的一套传输文件的通信标准，**FTP**规定了其行为规范和接口交换信息的集合。
- 使用者可以通过**FTP**程序来下载或上传文件。它的任务是通过网络将文件从一台计算机上传送到另一台计算机上，并保证其传输的可靠性，这就像在操作系统下将文件在本机磁盘之间复制一样。
- 属于应用层的协议。



匿名FTP

- 在Internet上要连接FTP服务器，大多要经过一个登录（Login）的过程，要求输入用户在该主机上登记的账号和密码。
- 若提供匿名（anonymous）FTP的服务，用户不需要主机的账号和密码即可进入FTP服务器。
- 使用匿名进入服务器时，通常只能浏览及下载文件，不能上传文件或修改服务器上的文件。但也有的服务器会提供一些目录供用户上传文件。



FTP的基本工作原理

- 在两个主机之间传送文件好像是很简单的事情。其实这往往是非常困难的。原因是众多的计算机厂商研制出的文件系统多达数百种，且差别很大。经常遇到的问题：
 - (1) 计算机存储数据的格式不同；
 - (2) 文件的目录结构和文件命名的规定不同；
 - (3) 对于相同的文件存取功能，操作系统使用的命令不同；
 - (4) 访问控制方法不同。



FTP的基本工作原理

- 文件传送协议**FTP**只提供文件传送的一些基本的服务，它使用可靠的**TCP**运输服务。
- **FTP**的主要功能是减少或消除在不同操作系统下处理文件的不兼容性。
- **FTP**使用客户/服务器方式。一个**FTP**服务器进程可同时为多个客户进程提供服务。**FTP**的服务器进程由两大部分组成：一个**主进程**，负责**接受新的请求**；另外有若干个**从属进程**，负责**处理单个请求**。



FTP的基本工作原理

- 主进程的工作步骤如下：

（1）打开熟知端口（端口号为**21**），使客户进程能够连接上。

（2）等待客户进程发出连接请求。



FTP的基本工作原理

- 主进程的工作步骤如下：

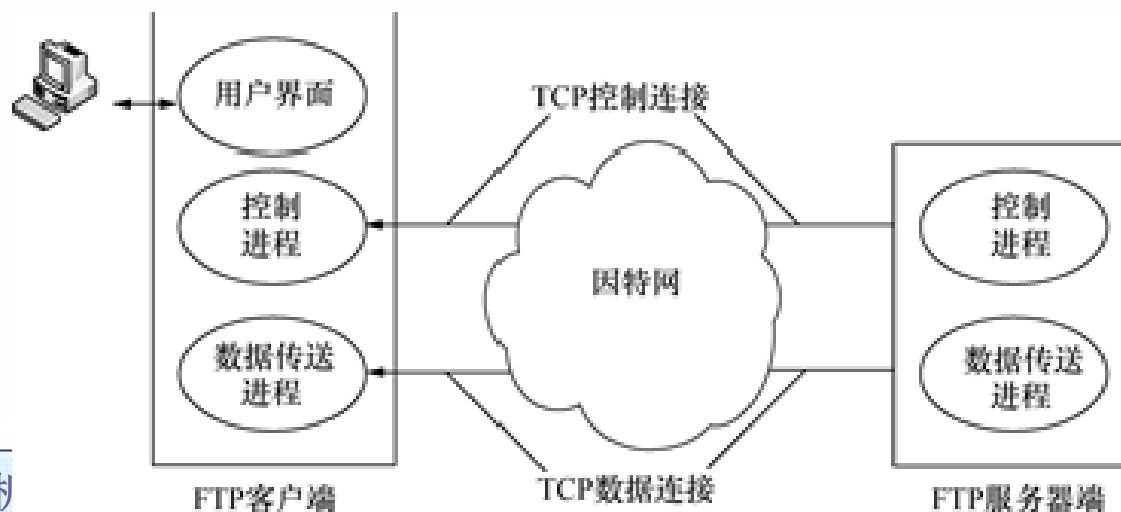
(3) 启动从属进程来处理客户进程发来的请求。从属进程对客户进程的请求处理完毕后即终止，但从属进程在运行期间根据需要还可能创建其他一些子进程。

(4) 回到等待状态，继续接受其他客户进程发来的请求。主进程与从属进程的处理是并发进行的。



FTP的基本工作原理

- **FTP**的工作情况如图所示。图中的圆圈表示在系统中运行的进程。图中的服务器端有两个从属进程：**控制进程**和**数据传送进程**。为简单起见，服务器端的主进程没有画上。在客户端除控制进程和数据传送进程外，还有一个**用户界面进程**作为和用户的接口。





FTP的基本工作原理

- 在进行文件传输时，FTP的客户和服务端之间要建立两个连接：“控制连接”和“数据连接”。
 - ⑩ 控制连接在整个会话期间一直保持打开，FTP客户所发出的传送请求通过控制连接发送给服务器端的控制进程。
 - ⑩ 控制连接并不用来传送文件。实际用于传输文件的是“数据连接”。



FTP的基本工作原理

- 服务器端的控制进程在接收到FTP客户发送来的文件传输请求后就创建“数据传送进程”和“数据连接”。
- 数据连接用来连接客户端和服务器的数据传送进程，数据传送进程实际完成文件的传送。
- 在传送完毕后关闭“数据连接”，并结束运行。



FTP的基本工作原理

- 当客户进程向服务器进程发出建立连接请求时，要寻找连接服务器进程的熟知端口，同时还要告诉服务器进程自己的端口号码，用于建立数据传送连接。
- 服务器进程用传送数据的端口与客户进程所提供的端口号码建立数据传送连接。



简单文件传送协议

- 简单文件传送协议（**Trivial File Transfer Protocol, TFTP**）是一个很小且易于实现的文件传送协议。
- **TFTP** 使用**C/S方式和UDP**数据报传输数据，因此 **TFTP** 需要有自己的差错改正措施。
- **TFTP** 只**支持文件传输而不支持交互**。**TFTP** 没有一个庞大的命令集，没有列目录的功能，也不能对用户进行身份鉴别。



简单文件传送协议

- TFTP的主要优点有两个:

- ⑩ 第一，TFTP可用于UDP环境;

- ⑩ 第二，TFTP代码所占的内存较小，减少了开销。



简单文件传送协议

● TFTP 的主要特点如下：

- ⑩ (1) 每次传送的数据 PDU 中有 512B 的数据，但最后一次可不足 512B。
- ⑩ (2) 数据 PDU 也称为文件块 (block)，每个块按序编号，从 1 开始。
- ⑩ (3) 支持 ASCII 码或二进制传送。
- ⑩ (4) 可对文件进行读或写。
- ⑩ (5) 使用很简单的首部。



简单文件传送协议

- **TFTP 的工作很像停止等待协议**，在发送完一个文件块后就等待对方的确认，确认时应指明所确认的块编号。
 - ⑩ 发完数据后在规定时间内**收不到确认就要重发数据 PDU**。
 - ⑩ 发送确认**PDU**的一方若在规定时间内收不到下一个文件块，也要重发确认**PDU**。
 - ⑩ 这样就可保证文件的传送不致于因某一个数据报的丢失而告失败。



简单文件传送协议

- 在一开始工作时，**TFTP 客户进程**发送一个读请求**PDU**或写请求 **PDU**给 **TFTP 服务器进程**，其熟知端口号码为 **69**。
- **TFTP 服务器进程**要选择一个新的端口和 **TFTP 客户进程**进行通信。若文件长度恰好为512B的整数倍，则在文件传送完毕后，还必须在最后发送一个只含首部而无数据的数据 **PDU**。若文件长度不是512B的整数倍，则最后传送数据 **PDU** 的数据字段一定不满**512B**，这正好可作为文件结束的标志。



7.6 远程终端协议

远程终端协议TELNET

- **TELNET**是一个简单的远程终端协议，也是因特网的正式标准。
- 用户用**TELNET**就可在其所在地通过**TCP**连接注册（即登录）到远地的另一个主机上（使用主机名或**IP**地址）。
- **TELNET**能将用户的击键传到远地主机，同时也能将远地主机的输出通过**TCP**连接返回到用户屏幕。这种服务是透明的，因为用户感觉到好像键盘和显示器是直接连在远地主机上。



远程终端协议

- **TELNET**也使用**C/S**方式。在本地系统运行**TELNET**客户进程，而在远地主机运行**TELNET**服务器进程。
 - ⑩和**FTP**的情况相似，服务器中的主进程等待新的请求，并产生从属进程来处理每一个连接。
- **TELNET**使用网络虚拟终端**NVT**格式，这种格式是为了适应许多计算机和操作系统的差异。例如，对于文本中一行的结束，有的系统使用回车（**CR**），而有得系统使用换行（**LF**）。



TELNET的工作原理

远程终端协议

● TELNET的工作原理:

- ⑩ 客户软件把用户的击键和命令转换成**NVT**格式，并送交服务器；
- ⑩ 服务器软件把收到的数据和命令，从**NVT**格式转换成远地系统所需的格式；
- ⑩ 向用户返回数据时，服务器把远地系统的格式转换为**NVT**格式；
- ⑩ 本地客户再从**NVT**格式转换到本地系统所需的格式。

104



7.7 简单网络管理协议

网络管理的基本概念

- 网络管理包括对**硬件、软件和人**力的使用、综合与协调，以便对网络资源进行监视、测试、配置、分析、评价和控制，这样就能以合理的价格满足网络的一些需求，如实时运行性能、服务质量等。



网络管理的需求

- (1) 网络**规模快速膨胀**，网络节点的大幅度增加、跨地域，这些造成网络需要集中维护。
- (2) 网络**复杂性大幅度增加**，网络设备种类多、应用种类多、网络协议种类多。
- (3) 网络运行的**可靠性要求**，通过配置冗余的网络硬件、软件、传输介质等。
- (4) 网络**运行可用性的要求**，降低网络出故障的时间，减少网络故障的修复时间。



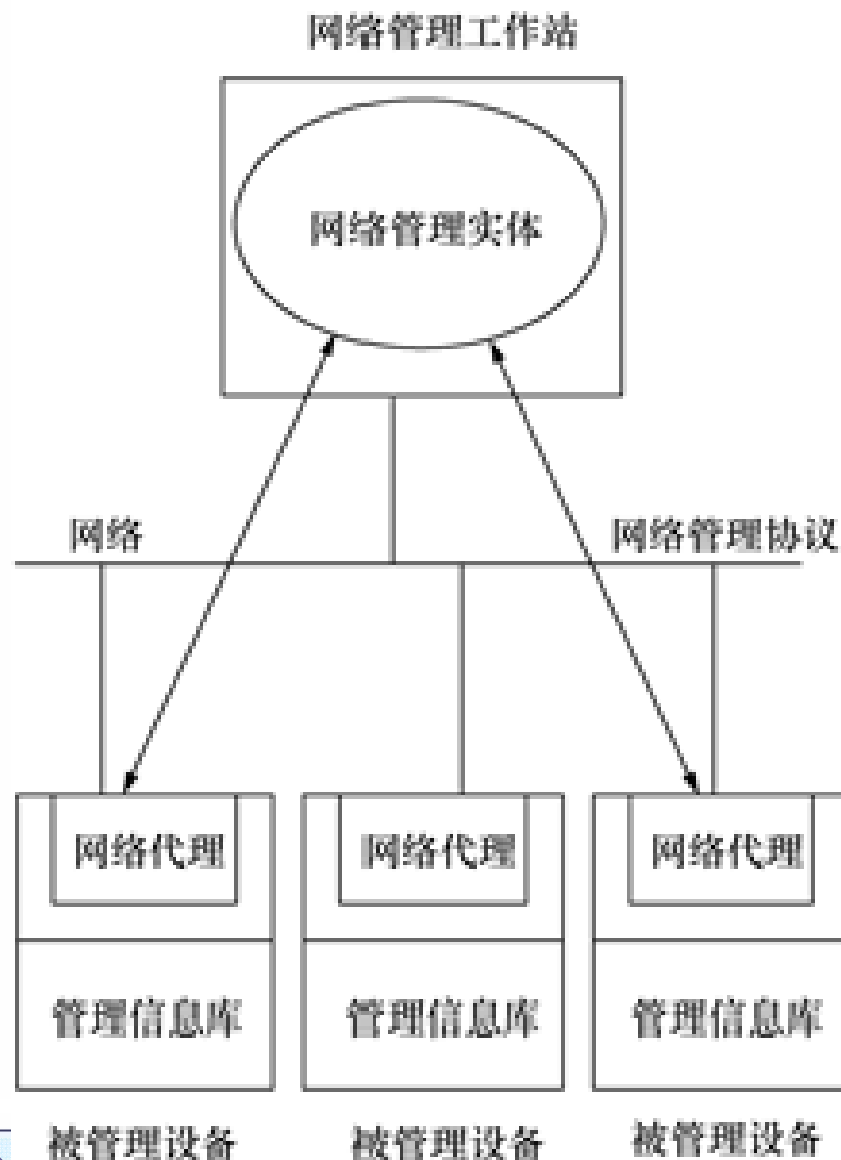
网络管理的需求

- (5) 网络**稳定性**的要求，要求提供一个运行平稳、网络性能有保障的网络。
- (6) 降低网络**运行维护成本**，集中维护模式的采用。
- (7) 网络**扩容、规划**的需要。

网络管理的**根本目标**就是为了满足运营者和用户对网络的有效性、可靠性、开放性、综合性、安全性和经济性要求。



网络管理中的主要构件





网络管理中的主要构件

● (1) 网络管理站

- 网络管理站由图形显示工作站和网络管理应用组成。
- 图形显示工作站为网络管理应用程序的运行提供存储、计算、通信和显示服务。
- 网络管理应用主要用来处理大量的数据、对事件进行响应、提供良好的人机接口界面。



网络管理中的主要构件

● (2) 被管理设备

- 被管理的设备可以是网络上的任何节点，例如，计算机、打印机、路由器、网络应用等；
- 被管理的设备通常有一个管理代理在后台运行。

● (3) 管理代理

- 管理代理向网络管理应用提供被管理设备的信息，同时也可以接收来自网络管理应用的控制信息。



网络管理中的主要构件

- (4) 网络管理协议

→网络管理协议用于网络管理应用和网络管理之间交换信息，定义网络管理信息的结构、数据格式等。

- (5) 网络管理信息

→网络管理信息是网络管理应用和网络管理代理之间交换的信息，用于监视和控制被管理的设备。



网络管理的功能

网络管理的功能

- (1) 性能管理：用于衡量和呈现网络特性的各个方面
 - 收集网络管理者感兴趣的那些变量的性能数据。
 - 分析这些数据，以判断是否处于正常（基线）水平并产生相应的报告。
 - 为每个重要的变量确定一个合适的性能阈值，超过该阈值就意味着出现了应该注意的网络故障。



网络管理的功能

- (2) 故障管理：用于检测、记录网络故障并通知给用户，尽可能自动修复网络故障以使网络能有效地运行。
 - 检测被管理对象的差错或接收差错事件报告。
 - 在紧急情况下启用备用资源用于服务。
 - 创建和维护差错日志库并对其进行分析。
 - 对故障进行诊断测试、跟踪、识别和定位。
 - 对有故障资源进行更换、修复，或采取其他恢复措施使其重新投入使用。



网络管理的功能

- (3) 配置管理：用于监视网络和系统的配置信息，以便跟踪和管理不同的软硬件元素对网络操作的作用
 - 网络资源的配置及其活动状态的监视
 - 网络资源之间的关系的监视与控制
 - 加入新资源，删除旧资源，定义新的管理对象
 - 识别管理对象，给每个对象分配名字
 - 初始化对象，启动、关闭对象
 - 管理各个对象之间的关系
 - 改变管理对象的参数



网络管理的功能

- (4) 计费管理：用于对用户使用网络资源的情况进行记录，并根据预定的计算方法和收费标准来核算用户的通信费用。
 - 统计网络效益数据，以便确定不同时期和不同时段的费用率。
 - 根据用户使用的特定业务和计费方式，确定和计算各个用户的费用。
 - 账务管理、分类和稽查功能。



网络管理的功能

- (5) **安全管理**: 用于控制对网络资源的访问, 以保证网络不被有意或无意地侵害, 并保证敏感信息不被那些未授权的用户访问。
 - ➔ 标识重要的网络资源 (包括系统、文件和其他实体)。
 - ➔ 确定重要的网络资源和用户集间的映射关系。
 - ➔ 监视对重要网络资源的访问。
 - ➔ 记录对重要网络资源的非法访问。



简单网络管理协议SNMP

SNMP

- 若要管理某个对象，就必然会给该对象添加一些软件或硬件，但这种“添加”必须对原有对象的影响尽量小些。
- 简单网络管理协议**SNMP**中的管理程序和代理程序按**C/S**方式工作。
- **SNMP**发布于**1988**年，**IETF**在**1990**年将**SNMP**定为因特网的正式标准，以后又有了新版本**SNMP v2**和**SNMP v3**，因此原来的**SNMP**又称为**SNMP v1**。



SNMP

- **SNMP最重要的指导思想就是要尽可能简单。**
- **SNMP的基本功能包括监视网络性能、检测分析网络差错和配置网络设备等。**
- **在网络正常工作时，SNMP可实现统计、配置和测试等功能；当网络出故障时，可实现各种差错检测和恢复功能。**
- **虽然SNMP是在TCP/IP基础上的网络管理协议，但也可扩展到其他类型的网络设备上。**



SNMP的工作过程

SNMP的工作过程

- (1) 管理软件和代理软件开始正常运行，网络管理员把要管理的子网或者网元加入到网络管理系统中，网络管理软件根据被管理的设备和链路，描绘出整个网络结构。
- (2) 管理软件定期轮询各网元设备。
- (3) 当无法轮询某个网元设备，就修改网络拓扑图，并发出报警。



SNMP的工作过程

- (4) 管理软件也接收来自各网元发出的中断事件。
- (5) 管理软件可以根据网管人员的要求，向各网元设备发出查询、参数设备等命令。



SNMP的工作过程

- 在SNMP协议中，**get**指令用来获取对象信息，**set**指令用来设置对象属性，**trap**指令用来实现中断消息的传递。
- 在SNMP协议工作时
 - ➔ 代理设备监听**UDP端口161**进行通信：如果采用轮询方式；
 - ➔ 管理工作端监听**UDP端口162**进行通信：如果采用中断方式。



SNMP的协议数据单元和报文

- **SNMP v1**一共定义了**五种协议数据单元类型**。

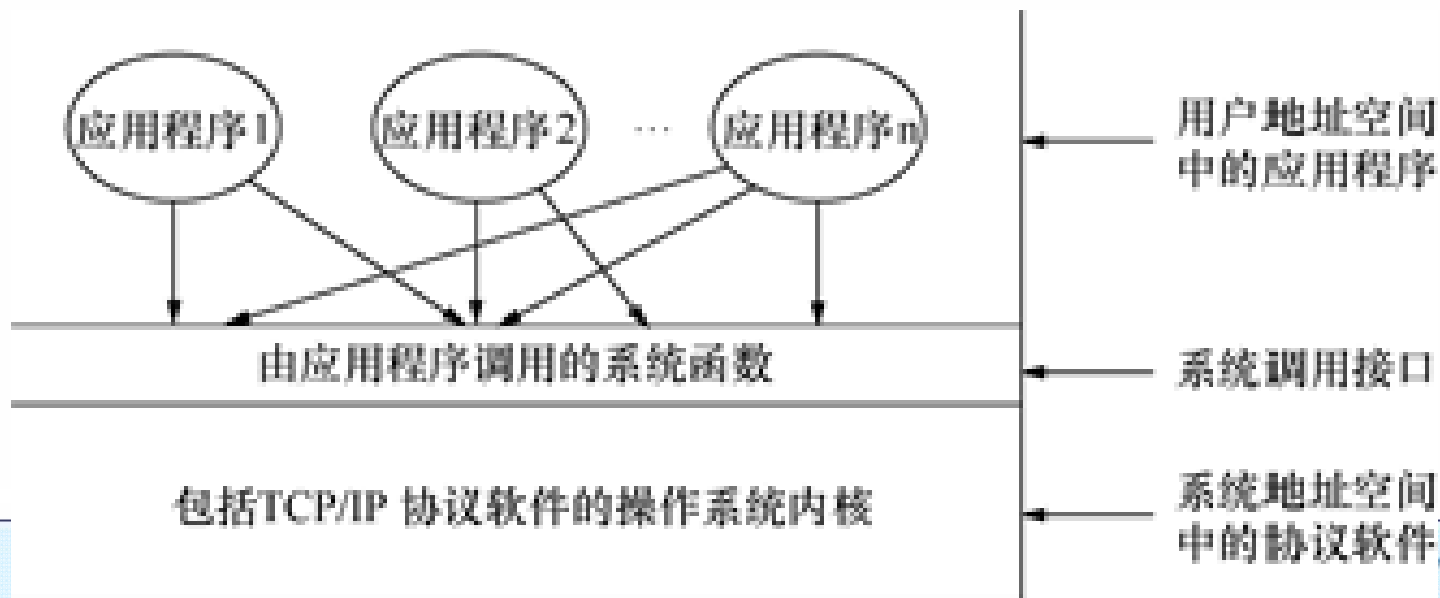
PDU 编号	PDU 名称	用 途
0	get-request	用来查询一个或多个变量的值
1	get-next -request	允许在一个 MIB 树上检索下一个变量
2	get-responses	对 get /set 报文作出响应
3	set-request	对一个或多个变量值进行设置
4	trap	向管理进程报告代理中发生的事件



7.8 应用进程跨越网络的通信

系统调用和应用编程接口

- 大多数操作系统使用系统调用（**system call**）的机制在应用程序和操作系统之间传递控制权。对程序员来说，每一个系统调用和一般程序设计中的函数调用非常相似，只是系统调用是将控制权传递给了操作系统。





系统调用和应用编程接口

系统调用和应用编程接口

- 当某个应用进程启动系统调用时，控制权就从应用进程传递给了系统调用接口。
- 此接口再将控制权传递给计算机的操作系统，而操作系统将此调用转给某个内部过程，并执行所请求的操作。
- 内部过程一旦执行完毕，控制权就又通过系统调用接口返回给应用进程。系统调用接口实际上就是应用进程的控制权和操作系统的控制权进行转换的一个接口，即应用编程接口（Application Programming Interface, API）。



系统调用和应用编程接口

- 目前只有几种可供应用程序使用的**TCP/IP**应用编程接口
 - **Berkeley UNIX** 操作系统中定义的一种**API**，它又称为插口接口（**socket interface**）；
 - 微软公司在其操作系统中采用了插口接口**API**，形成了一个稍有不同的**API**，并称之为**Windows Socket**。



系统调用和应用编程接口

- **Socket和API之间的区别:**

→ **Socket**是应用进程和网络之间的接口，因为**Socket**既包含运输层与应用层之间的端口号，又包含机器的**IP**地址。

→ **API**是应用程序和操作系统之间的接口。**API**从程序设计的角度定义了许多标准的系统调用函数。应用进程只要使用标准的系统调用函数就可得到操作系统的服务。



系统调用和应用编程接口

- 在Socket以上的进程是受应用程序控制的，而在Socket以下的TCP协议软件以及TCP使用的缓存和一些必要的变量等，则是受计算机操作系统的控制。
- 应用程序使用TCP/IP协议进行通信时，必须通过Socket与操作系统交互并请求其服务。应用程序的开发者和Socket以上的应用进程具有完全的控制，但对Socket以下的运输层却只有少量的控制。



几种常用的系统调用

- 主要的系统调用有以下几个

- ➔ (1) **socket()**调用：用来创建一个**socket**，这个调用返回一个整数。
- ➔ (2) **bind()**调用：指定**socket**所使用的**IP**地址和端口号，又称“本地插口地址”。
- ➔ (3) **recvfrom()**调用：将**socket**的入队列中的下一个数据报提取出来。



几种常用的系统调用

- 主要的系统调用有以下几个

- ➔ (4) **sendto()**调用：将一个数据报从出队列中取出，并用**UDP**发送给远地机器的一个进程。远地机器的**socket**地址是从上面的**recvfrom()**调用得到的。

- ➔ (5) **close()**调用：用来关闭一个**socket**。



几种常用的系统调用

- 主要的系统调用有以下几个

→ (6) **listen()** 调用：仅为**TCP**服务器使用的系统调用，其作用是使已经创建的**Socket**变成被动**Socket**，即监听**Socket**。监听**Socket**的用处不是和远地**Socket**建立连接，而只是等待远地客户发出的连接请求。

Listen()调用通知操作系统：服务器已做好接受连接的准备。



几种常用的系统调用

- 主要的系统调用有以下几个

- ➔ (7) **connect()**调用：通常是客户进程使用的系统调用，其功能是向远地进程（通常是服务器）请求建立连接。

- ➔ (8) **accept()**调用：是TCP服务器使用的系统调用，其作用是从入队列中提取最前面的连接要求。**accept()**调用会创建一个新的socket，叫做接受socket；此后，客户将只和这个新创建的接受socket建立连接并通信，而不再和监听socket发生联系。



几种常用的系统调用

- 主要的系统调用有以下几个

- ➔ (9) **fork()**调用：创建一个和自己完全一样的从属进程（或子进程）。
- ➔ (10) **read()**调用：读取从远地机器通过TCP连接传送到缓存中的数据。
- ➔ (11) **write()**调用：通过TCP连接将数据发送到远地机器的缓存中。



本章小结

- 内容

以Internet应用为例，介绍一些典型的应用层服务和协议，包括域名系统（DNS）、WWW（HTTP）、电子邮件（SMTP）等。

- 重点

域名系统（DNS）、WWW服务、HTTP协议、FTP、TELNET、SNMP、Socket