网络基础1:

1.网络的发展：单机 -> 网络

使用路由器/交换机这种设备将多个计算机连接起来实现数据交换--小型的网络

小型网络的互相连接数据交换----更大的网络

网络以覆盖范围划分:局域网/城域网/广域网　　互联网/因特网　　以太网/令牌环网--组网方式

2.在网络中必须能够唯一表示每一台主机，才能实现点到点的精确通信

IP地址: IPv4: uint32_ t　无符号4个字节的整数　　DHCP/NAT　　IPv6: uint8 _t addr[16];

网络通信中的每条数据都必须具备:源|P地址/目的IP地址--表示数据从哪个主机来,到哪个主机去

目的IP地址:能够让网络中的路由器为每一条数据根据 目的地址选择不同的路径到达对端主机

源IP地址:能够让对端主机知道数据是谁发送，以便于回复数据

3. IP地址使网络中实现主机与主机之间的通信,但是主机上有很多进程;通信中必须标识一条数据应该由哪个进程处理

端口: uint16_ t 无符号两个字节的整数 0~ 65535

一个进程可以使用多个端口，但是一个端口只能被一个进程占用

网络中的每条数据都必须具备:源端口/目的端口　　表示数据从哪个进程来, 到哪个进程去。

为什么不使用pid标识进程而是使用新的字段端口标识---进程的pid回随着程序的重启发生改变,但是端口不会

网络通信:不同主机之间的进程间通信;

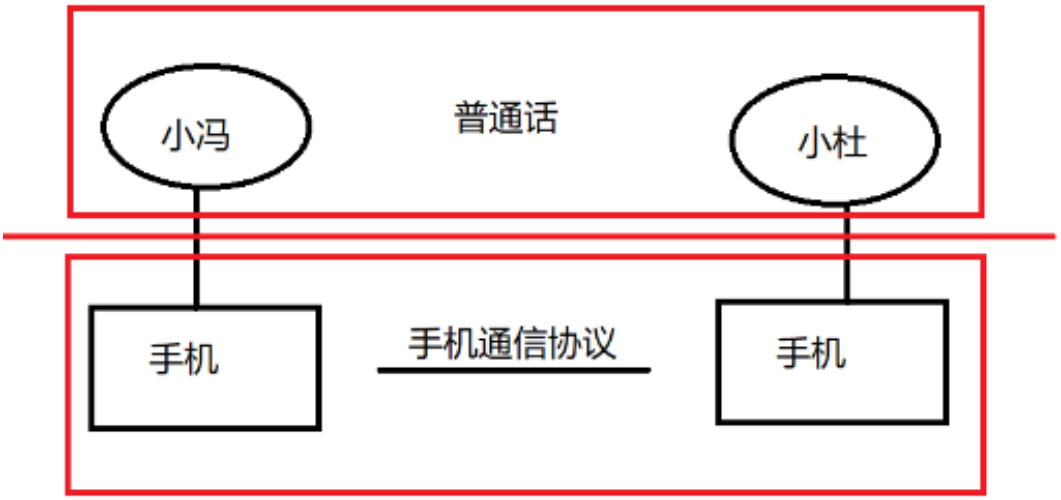
4.通过IP地址和端口可以实现不同主机之间的进程间通信了(网络通信)

协议:约定　　两个不同地区的人使用普通话通话- --普通话就是通信协议

网络通信协议:网络通信中数据格式的约定

协议分层:

对通信协议在不同的通信环境中进行封装，不同层次使用不同协议，提供不同的服务;将通信环境划分出来通信的实现更加简单，更容易形成规范



网络通信环境中的协议分层:

ISO: OSI七层参考模型: 应用层/表示层/会话层/传输层/网络层/链路层/物理层

TCP/IP五层模型:应用层/传输层/网络层/链路层/物理层

TCP/IP是一组协议栈/协议簇: 其中包含许多协议，IP和TCP协议只是其中比较典型的两种而已

应用层:负责应用程序之间的数据沟通;例如qq与qq之间的通信协议协商qq的数据格式;典型协议: HTTP/DNS/FTP

传输层:负责不同主机上进程间的数据传输;因为传输层的协议中包含主要信息就是端口;典型协议: TCP/UDP

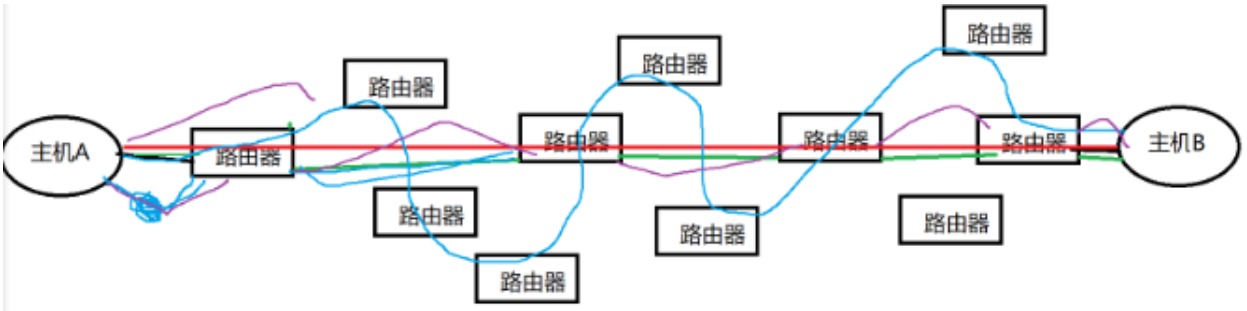
网络层:负责地址管理与路由选择;为网络中的数据选择合适的路径; IP信息; 典型协议: IP ;典型设备:路由器

链路层:负责相邻设备之间的数据帧识别以及传输;网卡设备的MAC地址信息;典型协议: Ethernet; 典型设备:交换机

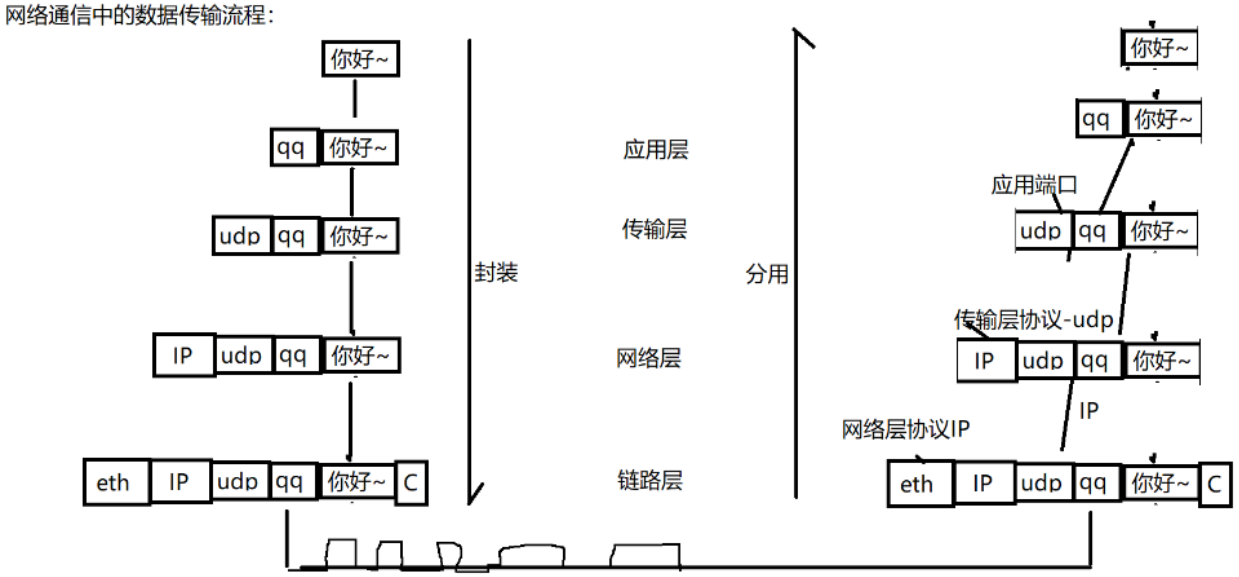
物理层:负责物理光电信号的传输;典型协议:以太网协议; 典型设备:集线器

协议分层:根据提供的服务不同,分出不同的通信层次,在每个层次使用不同的协议实现数据格式约定--将复杂的网络通信环境简化清晰

各层的典型设备和典型协议目前没有理解，就是记忆



网络通信中的数据传输流程：



网络字节序:

字节序: cpu在内存中对数据进行存取的顺序--取决于cpu架构

cpu架构: x86架构cpu---小端/ mips架构cpu---大端

大端字节序:低地址存高位　a[0]=01　a[1]=02　a[3]=03　a[4]=04

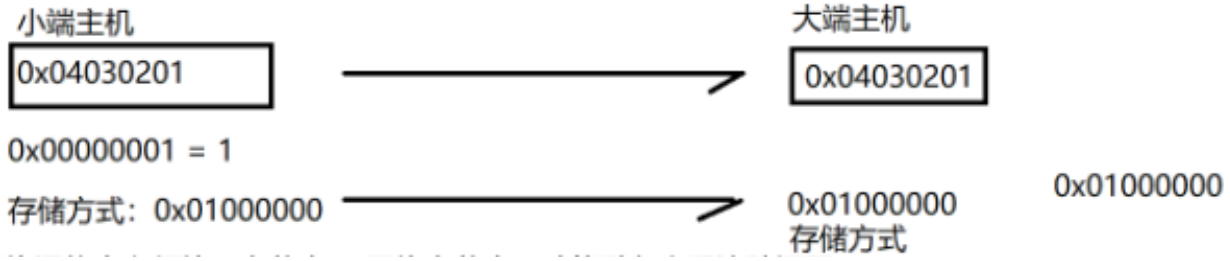
小端字节序:低地址存低位　a[0]=04　a[1]=03　a[2]=02　a[4]=01

0x01020304　　01是高位　　04是低位

uchar a[4]　　a[0] --低地址　　a[4]---高地址

主机字节序:当前主机的字节序

网络通信是不同主机之间的通信，但是不同主机上的主机字节序会对通信造成极大影响数据二义



因此网络通信中必须统一字节序--- 网络字节序,才能避免出现这种问题

不管你的主机是大端还是小端，网络通信的时候统-将数据转换为网络字节序---大端字节序

如果通信主机是一个小端主机，则在网络通信时需要将数据进行字节序转换才能发送

并不是所有的数据都需要转换:需要转换的数据关键 点在于内存中一次存取大小超过一个字节的数据: short int long float double

但是字符串char buf[1024]这是不需要转换的--字符串本身就是按字节存储的

需要转换的是小端主机，大端是不需要的--如何判断-个主机的字节序是大端还是小端

union tmp/int a; char　b (b就是a[0]) ;} tmp.a=1　　if (tmp.b == 1) {小端}

复习

网络发展背景:

以网络覆盖范围划分的网络:局域网/城域网/广域网

IP地址:

是什么: IPv4: uint32_ t --- 无符号4个字节的整数　DHCP/NAT;

IPv6: uint8_ t addr[16]

干什么:在网络中唯一标识一 台主机

注意:在网络中的每条数据中都需要包含两条信息:源IP/目的IP

PORT端口:

是什么: uint16_ t -- 无符号2个字节的整数

干什么:在一台主机上唯一标识一个进程

(一个主机上的程序运行起来之后，告诉操作系统自己使用哪个地址和端口,这时候操作系统从网卡上接收到数据之后，就会根据数据中的目的地址信息，选择这个数据交给哪一个进程来处理)

(发送端在发送数据的时候，就需要直接在数据中描述，这个数据是发给谁的，有个目的IP地址和端口信息)

注意:在网络中的每条数据中不仅需要包含IP地址信息，还需要包含:源端口/目的端口

特性:一个端口只能被一个进程占用，一个进程可以使用多个端口

协议:约定

网络通信中的协议:数据格式的约定

协议分层:

OSI七层参考模型:应用层/表示层/会话层/传输层/网络层/链路层/物理层

TCP/IP五层/四层模型:应用层/传输层/网络层/链路层/物理层

应用层:负责应用程序之间的数据沟通; HTTP　　描述应用数据如何组织解析以及描述信息

传输层:负责应用程序之间的数据传输; TCP/UDP　　描述端口信息标识从哪个进程到哪个进程

网络层:负责地址管理与路由选择; IP; 路由器　　描述IP信息标识从哪个主机到哪个主机

链路层:负责相邻设备之间的数据帧识别与传输;以太网协议:交换机　　描述MAC信息标识从哪个设备到哪个设备

物理层:负责物理光电信号的传输;以太网协议;集线器

数据的传输流程:

发送端主机的层层数据封装: tcp/ip模型中的五层，层层对原始数据进行描述
接收端主机的层层数据分用: tcp/ip模型中的五成，层层对网络数据进行解析得到原始数据的过程
网络字节序:
字节序: cpu对内存中的数据进行存取的顺序,取决于cpu的架构; x86-小端MIPS-大端
字节序分类:
大端字节序:低地址存高位
小端字节序:低地址存低位
低地址和高地址:指的是一个数据在内存空间中使用的地址是从低到高的　int a; a变量 的起始地址就是最小地址
低位和高位:指的是一个数据的二进制位从右到左，是越来越高的
网络字节序:为了防止网络通信中因为两端主机的主机字节序不同而造成数据二义所订立的标准
不管两端通信主机是大端还是小端，反正在网络通信的时候都要讲数据转换为网络字节序然后再进行传输
接收端接收到数据之后，从网络字节序转换为主机字节序再进行处理
网络字节序就是大端字节序; .
并不是所有的数据都需要转换，数据存储单位大于一个字节的数据:
short/int/long/float/double
char buf[1024]字符串是按字节为存储单位，因此字符串不涉及字节序转换
主机字节序如何在代码中判断:　union tmp{int a; uchar b;}　tmp.a=1　if(tmp.b= = 1) {是小端字节序}