

学习Linux网络编程之三 PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E5_AD_A6

[_E4_B9_A0Linu_c103_144240.htm](#) 2.7 总结 总的来说网络程序是由两个部分组成的--客户端和服务端.它们的建立步骤一般是: 服务端 socketbindlistenaccept 客户端 socketconnect 3. 服务器和客户机的信息函数 这一章我们来学习转换和网络方面的信息函数. 3.1 字节转换函数 在网络上面有着许多类型的机器,这些机器在表示数据的字节顺序是不同的, 比如i386芯片是低字节在内存地址的低端,高字节在高端,而alpha芯片却相反. 为了统一起来,在Linux下面,有专门的字节转换函数. unsigned long int htonl(unsigned long int hostlong) unsigned short int htons(unsigned short int hostshort) unsigned long int ntohl(unsigned long int netlong) unsigned short int ntohs(unsigned short int netshort) 在这四个转换函数中,h 代表host, n 代表network.s 代表short l 代表long 第一个函数的意义是将本机器上的long数据转化为网络上的long. 其他几个函数的意义也差不多. 3.2 IP和域名的转换 在网络上标志一台机器可以用IP或者是用域名.那么我们怎么去进行转换呢? struct hostent *gethostbyname(const char *hostname) struct hostent *gethostbyaddr(const char *addr,int len,int type) 在中 struct hostent的定义 struct hostent{ char *h_name. /* 主机的正式名称 */ char *h_aliases. /* 主机的别名 */ int h_addrtype. /* 主机的地址类型 AF_INET*/ int h_length. /* 主机的地址长度 对于IP4 是4字节32位*/ char **h_addr_list. /* 主机的IP地址列表 */ } #define h_addr h_addr_list[0] /* 主机的第一个IP地址*/ gethostbyname可

以将机器名(如 linux.yessun.com)转换为一个结构指针.在这个结构里面储存了域名的信息 gethostbyaddr可以将一个32位的IP地址(C0A80001)转换为结构指针.这两个函数失败时返回NULL且设置h_errno错误变量,调用h_strerror()可以得到详细的出错信息

3.3 字符串的IP和32位的IP转换.

在网络上面我们用的IP都是数字加点(192.168.0.1)构成的,而在struct in_addr结构中用的是32位的IP,我们上面那个32位IP(C0A80001)是192.168.0.1 为了转换我们可以使用下面两个函数

```
int inet_aton(const char *cp,struct in_addr *inp) char *inet_ntoa(struct in_addr in)
```

函数里面 a 代表 ascii n 代表network.第一个函数表示将a.b.c.d的IP转换为32位的IP,存储在 inp指针里面.第二个是将32位IP转换为a.b.c.d的格式.

3.4 服务信息函数

在网络程序里面我们有时候需要知道端口.IP和服务信息.这个时候我们可以使用以下几个函数

```
int getsockname(int sockfd,struct sockaddr *localaddr,int *addrlen) int getpeername(int sockfd,struct sockaddr *peeraddr, int *addrlen) struct servent *getservbyname(const char *servname,const char *protoname) struct servent *getservbyport(int port,const char *protoname) struct servent { char *s_name. /* 正式服务名 */ char **s_aliases. /* 别名列表 */ int s_port. /* 端口号 */ char *s_proto. /* 使用的协议 */ }
```

一般我们很少用这几个函数.对应客户端,当我们要得到连接的端口号时在connect调用成功后使用可得到系统分配的端口号.对于服务端,我们用INADDR_ANY填充后,为了得到连接的IP我们可以在accept调用成功后使用而得到IP地址.在网络上有许多的默认端口和服务,比如端口21对ftp80对应WWW.为了得到指定的端口号的服务我们可以调用第四个函数,相反为了得到端口号可以调用

第三个函数. 100Test 下载频道开通，各类考试题目直接下载。
详细请访问 www.100test.com