雪海茫茫 网络文摘

个人资料

cyx1743

访问： 20875次
积分： 276分
排名： 千里之外

原创： 4篇 转载： 19篇
译文： 0篇 评论： 1条

解析struct sockaddr_ll获得混杂模式

struct diagnostics internet session socket token

文章来源：http://hi.baidu.com/sjb811023/blog/item/bb0008635a16566a0c33fa22.html

参考：http://hi.baidu.com/zkheartboy/blog/item/3ce6c207000e10cf7a8947a0.html

测试过

sockaddr_ll, 源文件为<netpacket/packet.h>,结构如下：

```
struct sockaddr_ll
{
unsigned short int sll_family; /* 一般为AF_PACKET */
unsigned short int sll_protocol; /* 上层协议 */
```

```
int sll_ifindex; /* 接口类型 */
unsigned short int sll_hatype; /* 报头类型 */
unsigned char sll_pkttype; /* 包类型 */
unsigned char sll_halen; /* 地址长度 */
unsigned char sll_addr[8]; /* MAC地址 */
};
```

sll_family 和sockaddr_in中的sa_family一样，但选项要设置为AF_PACKET。
设置成这个选项后，从网卡接收的数据包可以直接传送到应用程序而不经过内核处理。

sll_protocol 表示上层的协议类型：
源文件为<linux/if_ether.h>

```
/*
* These are the defined Ethernet Protocol ID's.
*/

#define ETH_P_LOOP 0x0060 /* Ethernet Loopback packet */
#define ETH_P_PUP 0x0200 /* Xerox PUP packet */
#define ETH_P_PUPAT 0x0201 /* Xerox PUP Addr Trans packet */
#define ETH_P_IP 0x0800 /* Internet Protocol packet */
#define ETH_P_X25 0x0805 /* CCITT X.25 */
#define ETH_P_ARP 0x0806 /* Address Resolution packet */
#define ETH_P_BPQ 0x08FF /* G8BPQ AX.25 Ethernet Packet*/
#define ETH_P_IEEEPUP 0x0a00 /* Xerox IEEE802.3 PUP packet */
```

规范化driver的printk (372)
Fedora15 开启 TFTP 服 (324)

评论排行

用C语言实现Ping程序功 (1)
Vim+cscope+ctags+tags (0)
IP组播 (0)
select,poll,epoll用法 (0)
出现 warning: assignme (0)
SELECT设备超时用法小 (0)
转:vim插件 ctags 和 tagli (0)
Fedora15 开启 TFTP 服 (0)
Linux下获取网卡列表(/pr (0)
规范化driver的printk (0)

推荐文章

最新评论

用C语言实现Ping程序功能
xiaorongsmile4u: 解释的很详细

#define ETH_P_IEEEPUPAT 0x0a01 /* Xerox IEEE802.3 PUP Addr Trans packet */

#define ETH_P_DEC 0x6000 /* DEC Assigned proto */

#define ETH_P_DNA_DL 0x6001 /* DEC DNA Dump/Load */

#define ETH_P_DNA_RC 0x6002 /* DEC DNA Remote Console */

#define ETH_P_DNA_RT 0x6003 /* DEC DNA Routing */

#define ETH_P_LAT 0x6004 /* DEC LAT */

#define ETH_P_DIAG 0x6005 /* DEC Diagnostics */

#define ETH_P_CUST 0x6006 /* DEC Customer use */

#define ETH_P_SCA 0x6007 /* DEC Systems Comms Arch */

#define ETH_P_RARP 0x8035 /* Reverse Addr Res packet */

#define ETH_P_ATALK 0x809B /* Appletalk DDP */

#define ETH_P_AARP 0x80F3 /* Appletalk AARP */

#define ETH_P_IPX 0x8137 /* IPX over DIX */

#define ETH_P_IPV6 0x86DD /* IPv6 over bluebook */

#define ETH_P_PPP_DISC 0x8863 /* PPPoE discovery messages */

#define ETH_P_PPP_SES 0x8864 /* PPPoE session messages */

#define ETH_P_ATMMPOA 0x884c /* MultiProtocol Over ATM */

#define ETH_P_ATMFATE 0x8884 /* Frame-based ATM Transport

* over Ethernet

*/

一般是IP的话选ETH_P_IP。或者小心的选择ETH_P_ALL

sll_ifindex 表示接口类型，也可以选择：

源文件<linux/netdevice.h>

```
/* Media selection options. */
enum {
IF_PORT_UNKNOWN = 0,
IF_PORT_10BASE2,
IF_PORT_10BASET,
IF_PORT_AUI,
IF_PORT_100BASET,
IF_PORT_100BASETX,
IF_PORT_100BASEFX
};
```

但不知道真的起作用吗，我选的10BASET——基于双绞线的10M以太网。

AF_INET 在<bits/socket.h>里定义为2，表示IP protocol family.

AF_IENT和IF_PORT_10BASET的大小一样。

最好这样得到

```
1. int get_nic_index(int fd, const char* nic_name)
2. {
3.     struct ifreq ifr;
4.     if (nice_name == NULL)
5.         return -1;
6.     memset(&ifr, 0, sizeof(ifr));
7.     strncpy(ifr.ifr_name, nic_name, IFNAMSIZ);
8.     if (ioctl(fd, SIOCGIFINDEX, &ifr) == -1) {
9.         perro("SIOCGIFINDEX ioctl error");
0.         return -1;
```

```
    }
    return ifr.ifr_ifindex;
}
```

sll_hatype ARP 硬件地址类型

可以选择，

源文件为<net/if_arp.h>

```
/* ARP protocol HARDWARE identifiers. */
#define ARPHRD_NETROM 0 /* From KA9Q: NET/ROM pseudo. */
#define ARPHRD_ETHER 1 /* Ethernet 10/100Mbps. */
#define ARPHRD_EETHER 2 /* Experimental Ethernet. */
#define ARPHRD_AX25 3 /* AX.25 Level 2. */
#define ARPHRD_PRONET 4 /* PROnet token ring. */
#define ARPHRD_CHAOS 5 /* Chaosnet. */
#define ARPHRD_IEEE802 6 /* IEEE 802.2 Ethernet/TR/TB. */
#define ARPHRD_ARCNET 7 /* ARCnet. */
#define ARPHRD_APPLETLK 8 /* APPLEtalk. */
#define ARPHRD_DLCI 15 /* Frame Relay DLCI. */
#define ARPHRD_ATM 19 /* ATM. */
#define ARPHRD_METRICOM 23 /* Metricom STRIP (new IANA id). */
```

sll_pkttype 包含分组类型

有效的分组类型：

目标地址是本地主机的分组用的 PACKET_HOST，

物理层广播分组用的 PACKET_BROADCAST ，

发送到一个物理层多路广播地址的分组用的 PACKET_MULTICAST，
在混杂(promiscuous)模式下的设备驱动器发向其他主机的分组用的PACKET_OTHERHOST，
本源于本地主机的分组被环回到分组套接口用的 PACKET_OUTGOING。
这些类型只对接收到的分组有意义。sll_addr 和 sll_halen 包括物理层(例如 IEEE 802.3)地址和地址长度。精确的解释依赖于设备。

sll_halen 为MAC地址长度（6 bytes)
源文件<linux/if_ether.h>
#define ETH_ALEN 6 /* Octets in one ethernet addr */
#define ETH_HLEN 14 /* Total octets in header. */
#define ETH_ZLEN 60 /* Min. octets in frame sans FCS */
#define ETH_DATA_LEN 1500 /* Max. octets in payload */
#define ETH_FRAME_LEN 1514 /* Max. octets in frame sans FCS */

sll_addr[8] 为目的MAC地址（按sockaddr_in.sin_addr为目的IP地址推测）

填充完这个结构后，创建一个SOCKET
sockfd=socket(AF_PACKET,SOCK_RAW,htons(ETH_P_IP));

```
1. int set_nic_promisc(int sockfd, const char *nic_name)
2. {
3.     struct ifreq ethreq;
4.     strncpy(ethreq.ifr_name, nic_name, IFNAMSIZ);
5.     ioctl(sockfd, SIOCGIFFLAGS, &ethreq);
```

```
6.      ethreq.ifr_flags |= IFF_PROMISC;
7.      ioctl(sockfd, SIOCSIFFLAGS, &ethreq);
8.      return 0;
9.  }
```

如果以上步骤成功的话，网卡现在应该处于混杂模式了。

上一篇 利用原始套接字抓取数据

下一篇 内核模块相关命令：lsmod,depmod,modprob...

主题推荐 struct 应用程序 broadcast 以太网 socket

猜你在找

htons函数详解
Linux网络编程经典书籍推荐
MPI程序中的多进程写冲突问题的解决与遗留问题
嵌入式Linux如何查看硬件设备
(三)洞悉linux下的Netfilter&iptables：内核中的rule，match
libpcap使用
自动生成Makefile时，关于Makefile.am编写
LINUX下使用JsonCpp
2014木瓜移动校园招聘笔试题
轻松搞死VS2008的C++编译器

查看评论

暂无评论

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap