基于Linux的网络数据帧捕获方法与思考 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_9F_BA_E4_BA_8ELinu_c103_144371.htm 目前，国内推出了许多的Linux的发行版本，其重点集中在中文平台上，方便了国内用户对Linux的使用，但是有一个不是太好的迹象就是把汉化作为Linux操作系统的主要功能，实际上汉字处理尽管非常重要，但是把Linux作为桌面系统进行推广，其价值不是非常大，并且在推出的发行版本中，应用程序的源代码包多被删除，而选择了一些不是太有价值的X-Windows程序包，而许多应用程序（如PHP3）必须要源代码的支持才可进行功能的扩展，GNU/Linux的优秀主要是给了我们非常丰富的软件资源，和享受资源的充分自由，应用程序的分析难度远小于内核，并且能够带来比较明显的效果，实际上许多的应用程序都提供了多平台的支持。Linux目前可能作为对抗Windows NT的工具是非常合适的。 附源程序： /* * This program demonstrate SOCK_PACK call. * Thanks Linux. Thanks Alan Cox * derived from/usr/src/redhat/SOURCES/dosemu-0.66.7/src/dosext/net/net/libpacket.c * compile method： cc capturer.c -o capturer */ /* * Alan Cox raw code */ /* * SOCK_PACKET support. * Placed under the GNU LGPL. * * First cut at a library of handy support routines. Comments, additions * and bug fixes greatfully received. * * (c) 1994 Alan Cox iiitac@pyr.swan.ac.uk GW4PTS@GB7SWN */ #include #include #include #include #include #include #include #include #include #include #include #include /*#if __GLIBC__ > 1*/ #include #include /*#else #include #include #endif*/ #include #include /* *

Obtain a file handle on a raw ethernet type. In actual fact * you can also request the dummy types for AX.25 or 802.3 also * * -1 indicates an error * 0 or higher is a file descriptor which we have set non blocking * * WARNING: It is ok to listen to a service the system is using (eg arp) * but dont try and run a user mode stack on the same service or all * hell will break loose. */ int OpenNetworkType(unsigned short netid) { int s = socket(AF_INET, SOCK_PACKET, htons(netid)). if (s == -1) return -1. fcntl(s, F_SETFL, O_NDELAY). return s. } /* * Close a file handle to a raw packet type. */ void CloseNetworkLink(int sock) { close(sock). } /* * Write a packet to the network. You have to give a device to * this function. This is a device name (eg eth0 for the first * ethernet card). Please dont assume eth0, make it configurable * - plip is ethernet like but not eth0, ditto for the de600s. * * Return: -1 is an error * otherwise bytes written. */ int WriteToNetwork(int sock, const char *device, const char *data, int len) { struct sockaddr sa. sa.sa_family = AF_INET. strcpy(sa.sa_data, device). return (sendto(sock, data, len, 0, amp.sa, amp.req). close(s). /* Thanks Rob. for noticing this */ if (err == -1) return err. memcpy(addr, req.ifr_hwaddr.sa_data,8). return 0. } /* * Obtain the maximum packet size on an interface. * * Return: * >0 Return is the mtu of the interface * -1 Error. */ int GetDeviceMTU(char *device) { int s = socket(AF_INET, SOCK_DGRAM, 0). struct ifreq req. int err. strcpy(req.ifr_name, device). err = ioctl(s, SIOCGIFMTU, &amp.req). close(s). /* So Ill add this one as well. Ok Alan? - Rob */ if (err == -1) return err. return req.ifr_mtu. } #define

data_packet_len 1514 int main(int argc ,char *argv[]) { char devicename_rec[32]. unsigned char data[data_packet_len]. int netid=0x03,sock_h=0,i=0,count_rec=0. if ((sock_h=OpenNetworkType(netid))0) { printf("Received Packet = %d\n", count_rec) . for (i=0.i以上程序在Redhat 5.1下编译通过，运行良好。 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