

# 基于Linux串口的非接触式IC卡开发应用

姜远<sup>1</sup>, 高岭<sup>1</sup>, 杨建锋<sup>2</sup>

(1. 西北大学 信息学院, 陕西 西安 710069;

2. 西北大学 现代技术教育中心, 陕西 西安 710069)

**摘要:**在Linux操作系统平台下的串口通信基础上,研究了如何通过编写程序实现在Linux系统下,应用字符设备开发IC卡设备驱动模块。基于Linux串口编程和非接触式IC卡应用设计并实现了一个串口通讯过程,并以相应的程序实例加以说明。有效利用了非接触式IC卡的特点,提高了读写卡的速度。

**关键词:**Linux; 串行通讯; IC卡设备

**中图分类号:**TP393

**文献标识码:**A

**文章编号:**1673-629X(2008)01-0209-03

## Application of No-Contact IC Card Based on Serial Port of Linux OS

JIANG Yuan<sup>1</sup>, GAO Ling<sup>1</sup>, YANG Jian-feng<sup>2</sup>

(1. School of Information Science and Technology, Northwest University, Xi'an 710069, China;

(2. Centre of Modern Technology Education, Northwest University, Xi'an 710069, China)

**Abstract:** Presents the serial communication under Linux OS, and discusses how to realize program which to apply character equipment developing IC card driver module. Serial programming for Linux OS and apply of no-contact IC card. Design and implement a serial communication process and illustrate the corresponding program. Effective use of the characteristics of no-contact IC card, Improve the speed of read and write card.

**Key words:** Linux; serial communication; IC card equipment

## 0 引言

随着信息技术的快速发展,个人数据将成为未来信息的主要角色。IC卡作为卡基应用系统中的一种卡型,利用安装在卡中的集成电路(IC)来记录和传递信息,是一种将个人信息有效地送入信息网络并获得所需结果的有效办法。IC卡芯片具有写入数据和存储数据的能力,IC卡存储器中的内容根据需要可以有条件地供外部读取,或供内部信息处理和判定之用。

作为IC卡与读写设备数据交换的一种方式,非接触型IC卡是通过无线电波或电磁场感应的方式,将卡中集成电路内的数据与外部设备接口设备通信,卡片不用直接接触接口设备的电极就可以进行数据读写。通信距离可以从几厘米至几十米远。其主要优点是环境适应性强、可全天候、无接触地完成自动识别、跟踪与管理。

## 1 Linux中的串口通信设备模块

### 1.1 Linux设备文件

在Linux中硬件设备只是一个设备文件,应用程序可以像操作普通文件一样对硬件设备进行操作。这些设备文件提供了到大多数物理设备的接口,它们分为字符设备、块设备和网络设备。

(1)字符设备:在存取时没有缓存的设备,不使用系统缓冲,对设备文件的操作直接反映到硬件上,串口就属于字符设备。

(2)块设备:读写都有缓存的支持,并且能够随机存取。主要包括软盘设备、CD-ROM等。

(3)网络设备:任何网络设备都要经过一个网络接口即一个能够和其它主机交换数据的设备,通常接口是个硬件设备但也可能是个纯软件设备。

### 1.2 串口通信的设置

对串口硬件设备进行操作,有两种方法:一是利用Linux内核自带的串口驱动程序;另外一种方法就是直接读写串口硬件端口,在这里只使用第一种方法。

要设置串口属性,首先应在程序中包含termios.h的头文件,这个头文件包含终端控制结构和控制函数。

收稿日期:2007-03-06

基金项目:陕西省自然科学基金项目(2005f36)

作者简介:姜远(1978-),男,陕西西安人,硕士研究生,研究方向为网络管理;高岭,教授,研究方向为网络测量和网络管理。



其中有一个非常重要的数据结构 struct termios, 这个数据结构包含了所有的串口参数, 如波特率、数据位、输入输出方式等。设置串口属性就是对这个数据结构赋值。可以通过 termios.h 中的函数 tssetattr() 设置串口的 termios 参数; 通过函数 tsgetattr() 获取串口的 termios 参数。该数据结构的各成员的属性值以及参数含义可以参考文献[1]。

设置完通信参数后就可用 read 从串口读取数据和 write 向串口写数据了, 其实质就是对串口文件进行读写。运行程序时要注意用户是否有对要操作的串口文件进行读写的权力, 可以用 chmod 命令进行文件权限修改。

## 2 非接触式 IC 卡简介

Mifare 1 IC 智能(射频)卡的核心是 Philips 公司的 Mifare 1 IC S50。内建有高速的 CMOS EEPROM MCU 等。卡片上除了 IC 微晶片及一副高效率天线外无任何其他元件, 无源(无任何电池), 工作时的电源能量由卡片读写器天线发送无线电载波信号耦合到卡片上天线而产生电能, 一般可达 2V 以上供卡片上 IC 工作频率 13.56MHz。

Mifare 1 IC 卡上具有先进的数据通信加密并双向验证密码系统; 且具有防重叠功能, 能在同一时间处理重叠在卡片读写器天线的有效工作距离内的多张重叠的卡片。其通讯原理如图 1 所示。

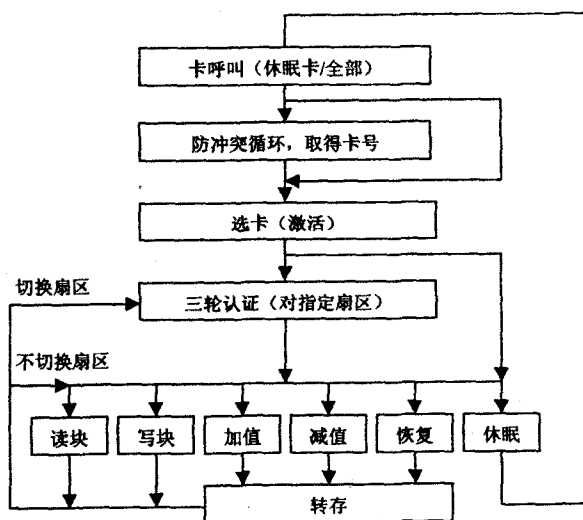


图 1 Mifare 1 IC 卡读写流程图

## 3 IC 卡设备在 Linux 下的开发实现

采取通用的 Mifare S50 卡为例, 详细描述其在校园机房的使用。

### 3.1 建立相应的数据结构

在头文件 AcctData.h 中, 定义结构体 AcctItem, 用

于反映机器的使用状态, 头文件 ICCard.h 中, 定义类 ICCard 用于反映 IC 卡的使用状态。确定在驱动模块程序中应用的数据结构的最终目的是读取和写入卡数据处理, 所以规范整齐的数据结构是必须的。

```

typedef struct {
    union {
        int ip; //ip 地址
        unsigned char b_ip[4];
    };
    union {
        char mac[16];
        unsigned long d_mac[4];
    }; //MAC 地址
    int id; //机器号
    int on_time; //刷卡上机时间
    int credit; //卡内余额
    int packets; //国际流量报数
    int bytes; //国际流量字节数
    int minutes; //上机时间
    int cost; //已花费金额数
    char icont[64]; //ic 卡中的密文
    int flag; //使用标记, 0:未使用 1:使用中 8:集体上机
} AcctItem;

class ICCard {
private:
    int fd, s_in, s_out; //套接字描述符, 读入数据量, 写入数据量
    struct termios oldtio, newtio; //串口结构定义 (使用前, 使用后)
    char * buf_in, * buf_out; //读入数据缓冲区首指针, 写入数据缓冲区首指针
    int Read(); //读取数据的处理
    int Write(int len); //写入数据的处理
    char Calc(char * buf); //相关密钥的处理
public:
    ICCard(const char * dev); //构造函数, 常量字符串标示了相应的串口号
    virtual ~ICCard();
    int ReadFail(char * buf); //读取数据失败的处理
    int WriteFail(char * buf); //写入数据失败的处理
    int Request(char, int); //搜索天线范围内是否有 Mifare 卡存在
    int Auth(char *); //进行读写器控制模块与 Mifare 卡的认证
    int Anticoll(char *); //通过防碰撞机制, 获得天线范围内一张卡的卡号
    int Select(char *); //选定天线范围内指定卡号的卡片, 进行读写处理
    int Config(char, char); //设置卡片与读写器模块之间的
  
```



通讯模式及率

```
};
```

这样,就定义了全部的卡处理数据结构定义以及设备操作映射函数结构,综上所述,就实现了 IC 卡设备的打开、读写和监控函数<sup>[2]</sup>。

### 3.2 模块初始化函数的实现

模块初始化函数是模块开发过程中必不可少的处理函数,用于实现设备的初始化,中断初始化及处理设备等等。

```
class PubData{
private:
    int GetCardData(char *, CardData &); //根据 ic 卡的密文读出用户数据
public:
    PubData();
    virtual ~PubData();
    init_waitqueue_head(&icdev, readq); //队列数据
    int GetFlag(int); //根据机器号获取使用标记
    int Online(char *, AcctItem &); //上机处理
    int Offline(AcctItem &, int n_tag=1); //下机处理
    int Resume(AcctData &); //故障恢复处理
    int GetComputer(AcctItem &); //恢复时读出数据库中记录的机器情况
};
```

在上面类中首先应用 GetCardData(char \*, CardData &)实现了卡数据的初始化;然后定义了队列数据,再进行了上机处理以及下机处理,故障恢复处理;最后实现了恢复故障处理。

### 3.3 中断处理

模块采用了定时器中断,在每个定时器中断发生时对刷卡状况进行检测,如果在天线范围内检测到刷卡,根据使用标记进行判断,未使用则进行读卡操作;使用中则进行写卡操作,同时更新 IC 卡的状态数据<sup>[3]</sup>。

程序中的中断处理采用了信号量机制来实现中断的后续处理,其处理流程为:

```
char * temp = new char[64];
while(1){
    int t = ic.CheckCard(); //检测是否插卡
    if(! t){
        memset(temp, 64, 0);
        ic.Read(32, 32, temp);
        Lock(mutex);
        memcpy(CardInfo, temp, 64);
        Unlock(mutex);
        Signal(hasCardInfo);
        t = ic.CheckCard();
        while(! t){
```

```
Sleep(1);
        t = ic.CheckCard();
    };
    else Sleep(1);
}
```

### 3.4 注销模块处理

用于模块卸载时进行资源的释放,在关闭串口前,需要通过 tcsetattr(fd, TCSANOW, &oldtio);将串口参数恢复为原来的设置,最后通过 close 系统调用关闭串口<sup>[4]</sup>,如下所述:

```
Void Remove_dev(void)
{
    delete buf_in, buf_out;
    tcflush(fd, TCIOFLUSH);
    tcsetattr(fd, TCSANOW, &oldtio);
    close(fd);
}
```

### 3.5 设备的读、写和配置函数

主要是通过实现 IC 卡的各种操作,在先前的类 ICCard 中已经介绍,根据 Mifare 1 IC 的通讯原理,实现卡的初始化操作。

综上所述,卡驱动模块的基本实现原理是:当有刷卡操作发生时,由使用标记引发中断或进行读卡操作;或进行写卡操作,同时进行相应数据处理。从而实现有卡操作发生时马上进行卡数据的更新<sup>[5]</sup>。

## 4 模块的编译调试

编写相应的 makefile 文件:

```
CC=gcc
LIB=-lstdc++ -lpq -lpq++ -lpthread -lcrypt
LIBDIR=-L/usr/lib
SRC=PubData.cpp ICCard.cpp ICsVrThrd.cpp Output.cpp AcctData.cpp MainThrd.cpp
OBJS=PubData.o ICCard.o ICsVrThrd.o Output.o AcctData.o MainThrd.o
INCLUDE=-I/usr/include/pgsql -I/usr/include -I.
BINDIR=.../bin
all:pubacct
pubacct: $(OBJS)
$(CC) -g -o pubacct $(OBJS) $(LIBDIR) $(LIB)
$(OBJS):
%.o: %.cpp
$(CC) -c $*.cpp $(INCLUDE) -g -o $*.o -DSERVERPART
install: pubacct qrysrv expire
cp pubacct $(BINDIR)
```

(下转第 215 页)



过大,使系统的性能下降。采用分页技术可解决数据量过大带来的问题<sup>[5]</sup>。

具体可以对 ADO 的 Recordset 对象中的 PageSize, AbsolutePage, PageCount 三个属性的使用来实现分页显示技术。其中 PageSize 代表页记录数, AbsolutePage 为当前页, PageCount 代表页数。讨论将要显示的页码值,即可轻松实现搜索数据的分页功能,具体的流程如图 6 所示<sup>[6]</sup>。

### 3 结束语

在 ISG 开发平台上设计与实现基于位置的手机博客系统,用户既可以通过手机短信接受手机博客系统服务,也可以通过 Web 页面接受服务。本系统的特色:博客与其创建的位置相关,用户可以根据自己的位

置创建博客和基于位置的查询。

#### 参考文献:

- [1] 叶核亚,陈立,廖雷. Java2 程序设计实用教程[M]. 北京:电子工业出版社,2003:254-261.
- [2] 吴伟. GSM/UMTS 移动位置业务的技术与实现[J]. 电信网技术,2003(4):14-19.
- [3] 万彭,许志远,张俊峰. 3G 业务开发平台的构建[J]. 现代电信科技,2006(10):51-55.
- [4] 飞思科技产品研发中心. Java2 应用开发指南[M]. 北京:电子工业出版社,2003:287-320.
- [5] 蒋伟钢. 基于 ASP 的动态网站核心技术分析[J]. 福建电脑,2005(2):30-32.
- [6] 周慧瑶,许锦标,吴煜林,等. 基于 ASP 的酒店商务网站的构建[J]. 计算机技术与发展,2006,16(11):213-215.

(上接第 163 页)

图像进行样本训练以及识别,并给出了识别结果。下一步工作将在不同的智能手机(摩托罗拉、诺基亚、夏新等)上测试人脸识别算法,并测试其他人脸识别算法,以求得较高识别率和较快的识别效率。人脸识别技术是人体生物特征识别技术的一种,其与智能手机的结合应用将是未来的研究热点,相信在智能终端、PDA 等移动设备方面都有很广泛的应用前景。

#### 参考文献:

- [1] 陈章龙. 嵌入式技术与系统——Intel XScale 结构与开发[M]. 北京:北京航空航天大学出版社,2003.
- [2] Diamantaras K I, Kung S Y. Principal Component Neural Networks: Theory and Applications[M]. New York: Wiley, 1996:50-59.
- [3] Luo Lin, Zou Cairong, Yang Fengfan. Modified algorithm of

principal component analysis for face recognition[J]. 东南大学学报:英文版, 2006,22(1):26-30.

- [4] 王守觉. 通用神经网络硬件中神经元基本数学模型的讨论[J]. 电子学报,2001,29(5):66-72.
- [5] 孙卫琴. Java 面向对象编程[M]. 北京:电子工业出版社,2006.
- [6] Boggs W, Boggs M. UML 与 Rational Rose 2002 从入门到精通[M]. 邱仲潘译. 北京:电子工业出版社,2002.
- [7] ORL. The ORL face database at the AT&T Research Laboratory[EB/OL]. 2007-03-02. <http://www.cl.cam.ac.uk/Research/DTG/attarchive/facedatabase.html>.
- [8] 刘振,吴鹏,陈月辉. 基于 PCA 和神经网络的人脸识别[J]. 山东科学,2006,19(4):63-67.
- [9] 何国辉,甘俊英. PCA 类内平均脸法在人脸识别中的应用研究[J]. 计算机应用研究,2006(3):165-167.

(上接第 211 页)

```
chown root $(BINDIR)/pubacct
chmod u+s $(BINDIR)/pubacct
clean:
```

```
rm -f *.o pubacct
```

执行几个 make 命令,便可以实现驱动模块的动态编译。

### 5 结束语

文中笔者创新点:介绍了非接触式 IC 卡的概况和发展前景。有助于有一定 IC 卡经验、技术及具有一定的计算机软/硬件技术的研发者设计出诸多应用场合的产品。这里只是截取了 IC 卡应用的一个方向,以及简单的设计开发。在其它的高级应用中还可以加入密

钥保护等措施。

#### 参考文献:

- [1] 刘峥嵘,张智超,许振山. 嵌入式 Linux 应用开发详解[M]. 北京:机械工业出版社,2005.
- [2] 尤晋元. UNIX 环境高级编程[M]. 北京:机械工业出版社,2004.
- [3] Sweet M R. Serial Programming Guide for POSIX Operating Systems[M]. [s.l.]:O'Reilly & Associates,2002.
- [4] Baumann P H. Serial Programming HOWTO[M]. [s.l.]:O'Reilly & Associates,2000.
- [5] 龙飞,李晓帆,蔡志开,等. 一个利用多线程及重叠 I/O 实现的串口通信类[J]. 微机发展,2004,14(3):51-53.