CRC 校验及其软件实现

姚七栋,张春玉
(陕西财经职业技术学院　陕西 咸阳　712000)

摘　要:数据通信技术是计算机网络技术发展的基础,已经成为现代生活中必不可少的一部分。但通过通信信道传输的数据往往会有差错的产生,而且差错的产生是不可避免的,我们的任务是分析差错产生的原因与差错类型,研究检查是否出现差错及如何纠正差错。循环冗余码(CRC)是目前应用最广的检错纠错编码方法之一。论述了 CRC 的教学原理及其在数据通信中的作用,并提出了用 8031 汇编语言实现 CRC 校验的程序设计。

关键词:CRC 校验;信道;误码;检错
中图分类号:TN911　　**文献标识码**:B　　**文章编号**:1004-373X(2006)13-067-02

Test of CRC and Its Realization

YAO Qidong,ZHANG Chunyu
(Shaanxi Professional and Technology Institute of Finance and Economics,Xianyang,712000,China)

Abstract: The data communication technology is the basis of development of the Internet technology,and has became the necessary part of modern life. However,there are often some mistakes for the date transmitted by communication way,and these mistakes can not be avoided. Our task is to analyze reasons and types of the mistakes,to study and check whether there is mistake appearing and how to correct it. At present,CRC is one of the most often used method to test and correct mistakes. This paper discusses the teaching principle of CRC and its function in communication,and points out how to realize the procedure design of CRC test.

Keywords: CRC test;communication way;error code;check mistake

在计算机通信过程中,由于信道上存在的各种复杂因素(例如:冲击噪声和热噪声等)的影响,所传输的信号将受到不同程度的干扰,严重时会造成误码以致阻断通信。所以应该在接收方检查所接收的数据是否正确,可采用多种检测方法。其中循环冗余校验码(CRC 校验)是目前在计算机网络通信及存储器等方面应用最为广泛的一种校验编码方法,是一种强有力的检测手段。人们将该技术用于多处数据通信系统中,收到了令人满意的效果。本文从 CRC 校验的教学原理及汇编语言实现两方面入手,详细介绍了 CRC 校验方法。

1　CRC 校验的教学原理

CRC 检错方法的工作原理是:将要发送的数据比特序列当做一个多项式 $K(X)$ 的系数,在发送端用收发双方预先约定的生成多项式 $G(x)$ 去除,求得一个余数多项式 $R(x)$。将余数多项式加到数据多项式之后(称为 $T(X)$)发送到接收端。在接收端用同样的生成多项式 $G(x)$ 去除接收到的数据多项式 $T'(X)$,如果除得尽,表明无差错,即 $T(X) = T'(X)$;如果除不尽,表明有差错,即 $T(X) \neq T'(X)$;余数将指明出错位所在位置。CRC 是一种线性分组码,具有较强的纠错能力并有许多特殊的代数性质,前 k 位为信息码元,后 r 位为校验码元,他除了具有线性分组码的封闭性之外,还具有循环性。其编码和译码电路很容易用移位寄存器实现,因而在 FEC 系统中得到了广泛的应用。

现以图 1 为例来说明 CRC 校验的教学原理。

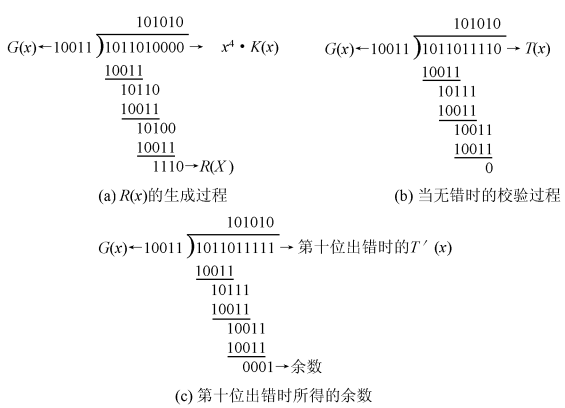

图 1　CRC 校验的教学原理

(1) 首先将欲传送的比特序列 $K(X)$ 乘以 X^r,其中 r 为 $R(x)$ 的位数,其值等于 $G(x)$ 的位数减 1。

收稿日期:2006-03-12

(2) 将乘得的结果 $X^r \cdot K(X)$ 用生成多项式 $G(x)$ 去除,忽略其商,仅将其余数 $R(x)$ 取出,并将其与 $X^r \cdot K(X)$ 相加,得到 $T(X)$,即 $T(X) = X^r \cdot K(X) + R(x)$。

(3) 需要指出的是:在 CRC 中,采用了一种以按位加减为基础的模 2 运算,不考虑进位和错位,即通过模 2 减实现模 2 除,以模 2 加将所得余数拼接在被除数后面,形成一个能除尽的校验码。模 2 加减即按位加减,相当于"异或",可用异或门硬件逻辑实现,当然也可用软件实现。举例如下:

①设 $K(X) = x^5 + x^3 + x^2 + 1$　　(101101)

$G(X) = x^4 + x + 1$　　(10011)

$G(X)$ 为 5 位,故 $R(X)$ 应为 4 位。

② $X^r \cdot K(X) = x^4(x^5 + x^3 + x^2 + 1)$

$= x^9 + x^7 + x^6 + x^4$　　(1011010000)

③算出 $X^r \cdot K(X)/G(X)$,取其余数:

$R(X) = x^3 + x^2 + x$　　(1110)

④将 $R(X)$ 与 $X^r K(X)$ 相加得到:

$T(X)$　　(1011011110)

⑤以上 $T(X)$ 即为将要实际传送的数据,图 1(b) 表示:用接收端收到的 $T'(X)$ ($T'(X) = T(X)$) 去除 $G(X)$,若除尽,则表示传输过程中无错。图 1(c) 表示:若第十位出错,则余数就不为 0。

⑥在接收端某位出错,则余数不为 0,不同位出错则余数不同,余数代码与出错位序号之间有惟一的对应关系。通过上例可求出其余数与出错位序号之间的对应模式(如表 1 所示),出错模式只与 CRC 码制和生成多项式有关,而与不同待传码字代码无关。表 1 对于(10,6)码具有一定的通用性,可作为其出错判别依据。

表 1　(10,6) CRC 循环码的出错模式(G(x) = 10011)

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	余数	出错位
正确的校验码	1	0	1	1	0	1	1	1	1	0	0000	无
接收端一位出错校验码	1	0	1	1	0	1	1	1	1	1	0001	10
	1	0	1	1	0	1	1	1	0	0	0010	9
	1	0	1	1	0	1	1	0	1	0	0100	8
	1	0	1	1	0	1	0	1	1	0	1000	7
	1	0	1	1	0	0	1	1	1	0	0011	6
	1	0	1	1	1	1	1	1	1	0	0110	5
	1	0	1	0	0	1	1	1	1	0	1100	4
	1	0	0	1	0	1	1	1	1	0	1011	3
	1	1	1	1	0	1	1	1	1	0	0101	2
	0	0	1	1	0	1	1	1	1	0	1010	1

⑦当然,对于其他码制或选用其他生成多项式,出错模式有可能不同。表 1 中列举了 11 种情况,一种是正确码字,除后余数为 0,其余 10 种是依次有位出错,余数不为 0,与出错位序号有惟一的对应模式。

⑧经过进一步的研究,我们发现了一个有实用价值的规律:如果有一位出错,用 $G(x)$ 除后得到一个不为 0 的余数,如果对该余数补 0,继续除,各次余数将按表 1 顺序循环,例如第十位出错,余数 0001,补 0 后继续除,得余数 0010,以后将依次为 0100,1000,0011,0110,1100,1011,0101,1010,0111,1110,1111,1101,1001,然后又是 0001,呈循环状。所以称之为"循环"码。

⑨循环冗余校验码的检错能力取决于生成多项式的选择,但并不是任何一个多项式都可以作为 $G(x)$,若从检错纠错的目的出发,生成多项式应能满足下列要求:任何一位数据发生错误时都应使余数不为 0,不同位出错则余数不同,余数代码与出错位序号之间最好有惟一的对应关系,并满足余数循环规律。

举例介绍几种生成多项式:

CRC-12,他具有 $G(x) = x^{12} + x^{11} + x^3 + x^2 + 1$ 的形式,由 12 位冗余位组成。

CRC-16,其形式为 $G(x) = x^{16} + x^{15} + x^2 + 1$,由16 位冗余位组成。

CRC-CCITT,其形式为 $G(x) = x^{16} + x^{12} + x^5 + 1$,由 16 位冗余位组成。

2　CRC 校验的汇编语言实现

在数据通信中,采用此校验方法,可使通信的误码率大为降低,确保了数据通信的可靠性。在程序编制过程中,高级语言的实现较为容易,在此仅给出 8031 汇编语言的打包与校验程序。若是对数据实行打包,可将数据的一组先增加 2 B 的长度,并将这 2 B 清零,经 CRC 计算后,形成 2 B 的冗余位;在接收方,则直接将此数据校验。程序中 R1 和 R2 即为冗余位的高 8 位和低 8 位,程序中的一些变量如 CNT_L,CNT_H 等可用 8031 的内部寄存器,原始数据存于 @DPTR 所指之处。

应该指出,我们所采用的通信方式通常为异步方式,而 CRC 属于同步方式的一种校验方法,关于这一点,可先将数据放入 RAM 中,把每一个存储单元的数据看成是同步传输数字序列的 8 个位来进行处理。CRC 校验程序如下:

```
CRC_16:NOP;                          ;CRC-16SUB
       MOV R3,#00H
       MOV R2,#00H
       MOV R1,#00H
CRC1:MOVX A,@DPTR
     INC DPTR
     MOV BYTE,#08H                   ;READ 1BYTE DATA
CRC2:CLR C
     RLC A                           ;C←MSB
     MOV A,R1
     RLC A
     MOV R1,A
     MOV A,R2
     RLC A
     MOV R2,A
     MOV A,R3
```

(下转第 71 页)

当用户进入系统时,必须在登陆主页输入用户名及口令,只有通过登陆验证的用户才能进行学习和管理。在系统中控制用户以同一登陆名连续非法登陆系统的次数,以防止攻击者进行连续性的操作从而猜测出该登陆名的密码。

(1) 口令加密认证

用户在对 Web 服务器进行存取时,可以通过对用户输入的口令进行认证,以确定用户的合法性。用户登录系统后,输入用户口令,服务器端将该口令用 MD5 算法进行加密,并将密文存放到用户的 Cookies 中。当用户再次登陆到该系统时,服务器要求用户输入口令,服务器端根据用户输入的口令与 Cookies 中的口令相比较,相同即为合法[7]。

(2) 数字签名认证

数字签名认证是采用 RSA 算法对用户进行认证。设服务器知道用户公钥,当用户访问该服务器时,用户端生成一个时戳,用私钥对时戳进行加密,并用本地的 Cookies 生成程序将密文生成一个相关的 Cookie。当用户登录到该服务器时,服务器取出该 Cookie 对其进行解密,并认证用户的身份。

(3) 分组管理

对用户进行分组管理。将用户分为安全级别不同的 3 个等级(系统管理员、教师、学生),每一等级的用户只能访问与其等级相应的系统资源和数据,控制用户可以访问哪些目录、子目录、文件和其他资料及用户可执行的操作。如果是教师或系统管理员,可采用与 IP 地址绑定,当用户登录时读取 IP 地址,以防止非法访问。

4 结 语

安全问题是任何 Web 程序开发所需关注的首要问题,NSI 系统需要对安全性问题给予足够的重视。文中仅对 NSI 系统的安全性作了初步探讨,对于要建造一个真正安全的系统是远远不够的,其安全性问题值得深入研究。

参 考 文 献

[1] 唐超,莫赞,冯珊. 面向远程教育的交互式教学及安全管理[J]. 计算机工程与应用,2002,38(17):218-220.

[2] 徐峥. 谈远程教育中电子邮件的安全防范[J]. 河南教育学院学报,2002,11(1):67-69.

[3] 唐汇国,张泰山,陈志盛. 基于 ASP 的数据库连接[J]. 计算机工程与应用,2003,39(31):191-194.

[4] 许锦波. Internet/ Intranet 网络安全结构设计[M]. 北京:清华大学出版社,2000.

[5] 冯登国,裴定一. 密码学导引[M]. 北京:科学出版社,1999.

[6] 李晓黎,张巍. ASP + SQL Server 网络应用系统开发与实例[M]. 北京:人民邮电出版社,2004.

[7] 沈洁,薛贵荣. 基于 Cookies 的分布式多 Web 系统的认证[J]. 计算机工程与应用,2002,38(18):151-153.

作者简介 徐 晶 女,1967 年出生,江苏扬州人,讲师,硕士。主要从事计算机软件与应用、计算机教育方向的研究。

(上接第 68 页)

```
        RLC A
        MOV R3,MOV A,R3
        JBC ACC.0,CRC5
CRC3:DEC BYTE
        MOV A,BYTE
        JNZ CRC6
        MOV A,CNT_L
        JZ CRC7
        DEC CNT_L
        MOV A,CNT_L
        JZ CRC4
        LJMP CRC1
CRC4:MOV A,CNT_H
        JZ CRC8
        LJM CRC1
CRC5:MOV A,R3
        XRL A,#01H                    ;MSB
        MOV R3,A
        MOV A,R2
        XRL A,#80H
        MOV R2,A
        MOV A,R1
        XRL A,#05H                    ;LSB
        MOV R1,A
        LJMP CRC3
CRC6:LJMP CRC2
CRC7:MOV A,CNT_H
        JZ CRC8
        DEC CNT_H
        MOV CNT_L,#0FFH
        LJMP CRC1
CRC8:RET
```

尽管 CRC 校验的原理相对简单,但由于不同系统之间存在性能要求、应用环境等差异,造成了 CRC 校验在实际上具有多种不同的形式,例如对编码和校验速度的要求常通过直接用硬件实现来解决,在没有条件使用硬件的情况下也可以设计快速算法利用软件来校验。对差错检验的精度则可通过采用不同长度的 CRC 编码算法来解决(例如 CRC-32 出错的概率比 CRC-16 要低出很多倍)。在一些要求较高的应用系统中,采用 32 位的 CRC 编码在众多差错检验方案中占据了主导地位。

参 考 文 献

[1] 雷建军. 计算机网络实用技术[M]. 北京:中国水利水电出版社,2001.

[2] 吴功宜,吴英. 计算机网络教程[M]. 北京:电子工业出版社,2003.

[3] Andrews,Tanenbaum. 计算机网络[M]. 3 版. 熊桂喜,王小虎,译. 北京:清华大学出版社,1998.