

Реализация алгоритма циклического избыточного кодирования с помощью языка C++

Колесников Алексей Александрович

*Приамурский государственный университет имени Шолом-Алейхема
студент*

Журавлева Ульяна Сергеевна

*Приамурский государственный университет имени Шолом-Алейхема
студент*

Лучанинов Дмитрий Васильевич

*Приамурский государственный университет имени Шолом-Алейхема
старший преподаватель кафедры информационных систем, математики и
методик обучения*

Аннотация

В данной статье рассмотрен алгоритм избыточного циклического кодирования. Описана реализация данного алгоритма с помощью языка C++.

Ключевые слова: C++, циклическое избыточное кодирование, полином, контрольная сумма.

C ++ cyclic redundancy coding algorithm implementation

Kolesnikov Aleksey Aleksandrovich

*Sholom-Aleichem Priamursky State University
student*

Zhuravleva Ul'jana Sergeevna

*Sholom-Aleichem Priamursky State University
student*

Luchaninov Dmitry Vasilyevich

*Sholom-Aleichem Priamursky State University
Senior lecturer of the Department of Information Systems, Mathematics and
teaching methods*

Abstract

In this article the algorithm of cyclic redundancy coding is considered. The realization of this algorithm using the C ++ language is described.

Keywords: C ++, CRC encoding, polynomial, check sum.

На сегодняшний день существует проблема устойчивой к помехам и ошибкам передача информации в информационной среде. Одним из способов контроля передачи данных является проверка контрольной суммы сообщения. Контрольная сумма – некоторое значение, рассчитанное по набору данных путём применения определённого алгоритма и используемое для проверки целостности данных при их передаче или хранении. Так как при передаче, данные могут исказиться, существует множество контролирующих передачу кодов. Одним из них является циклический избыточный код.

Алгоритм циклического избыточного кодирования был рассмотрен в статье Ross N. Williams, где было проведено полное описание кода и реализованы табличные алгоритмы их вычисления [1]. Также, объяснение работы циклического избыточного кодирования описано на сайте habrahabr, в статье «Простой расчет контрольной суммы» [2].

Целью исследования является один из кодов реализации группового кодирования – циклическое избыточное кодирование.

Кодирование на примере одного символа:

1. Предположим, что нужно закодировать букву 's'. Первым делом требуется привести её к двоичному представлению по таблице ASCII. Таким образом, буква примет вид '01110011'.

2. Следующим шагом, который надо сделать, это найти контрольную сумму символа. Для этого необходимо двоичную последовательность символа представить в виде полинома и делить на другую двоичную полиномиальную последовательность. Для деления в примере была взята последовательность '10011'. Таблица наиболее популярных полиномов представлена на сайте Википедия [3]. Найденной для символа 's' контрольной суммой является '1010'.

3. Последний шаг – это добавить к двоичной последовательности символа найденную для него контрольную сумму.

В итоге символ 's' будет закодирован в следующую последовательность: 011100111010.

Для реализации кода были разработаны следующие функции:

1. strhex
2. BinToInt
3. crc

Функция strhex переводит символ в его двоичное представление в таблице ASCII. Для этого средствами C++ получается номер символа в таблице ASCII и полученное число переводится в двоичную систему счисления.

```
string strhex (string str){
    string sum;
    int a=(int)str[0];
    while (a!=0){
        sum+=char(a%2)+48;
        a/=2;}
    while (sum.length()!=8) sum+='0';
```

```
reverse(sum.begin(), sum.end());
return sum;}
```

Функция BinToInt переводит символы из двоичной системы счисления в десятичную.

```
int BinToInt(string s){
    int i;
    int res = 0;
    int l = s.length();
    for (i = 0; i < l; i++){
        res *= 2;
        if (s[i] == '1') res += 1;}
    return(res);}
```

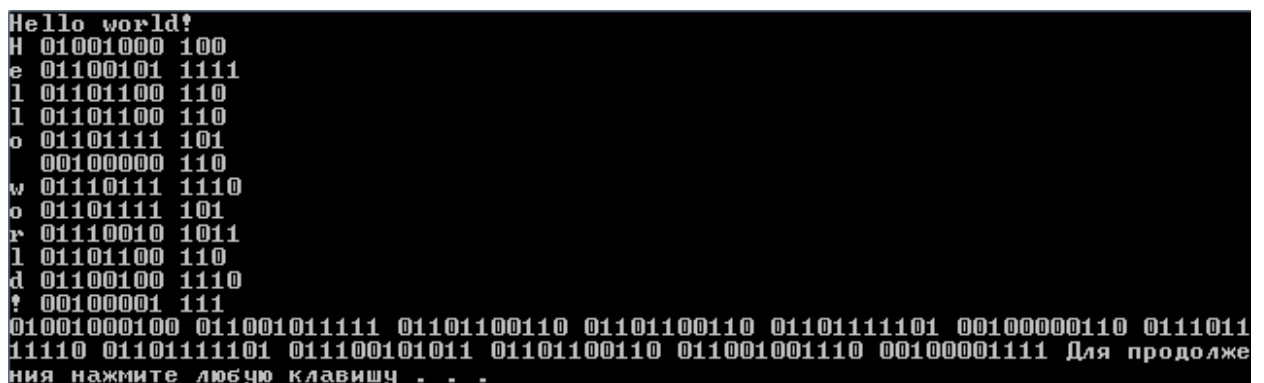
В функции crc выполняется процесс нахождения контрольной суммы

```
string crc (string str, string poly){
    string temp;
    while (str[0]=='0') str.erase(0,1);
    while (BinToInt(str)>BinToInt(poly)){
        for (int i=0;i<poly.length();i++){
            if (str[i]==poly[i]) str[i]='0'; else if
(str[i]!=poly[i]) str[i]='1';}
        while (str[0]=='0') str.erase(0,1);}
    while (str[0]=='0') str.erase(0,1);
    return str;}
```

В конце происходит применение функции crc. На вывод функции уходит уже двоичная последовательность символа с найденной и присоединенной контрольной суммой.

```
int main(){
    string a,t,s,p;
    getline(cin,a);
    p="10011";
    string *d=new string [a.length()];
    for (int i=0; i<a.length(); i++){
        t=a[i];
        d[i]=crc(strhex(t),p);
        s+=strhex(t)+d[i]+' ';
        cout<<t<<" "<<strhex(t)<<" "<<d[i]<<endl;}
    cout<<s;}
```

Пример работы алгоритма на примере фразы «Hello world!» (рис. 1).



```
Hello world!
H 01001000 100
e 01100101 1111
l 01101100 110
l 01101100 110
o 01101111 101
 00100000 110
w 01110111 1110
o 01101111 101
r 01110010 1011
l 01101100 110
d 01100100 1110
! 00100001 111
01001000100 011001011111 01101100110 01101100110 01101111101 00100000110 0111011
11110 01101111101 011100101011 01101100110 011001001110 00100001111 Для продолже
ния нажмите любую клавишу . . .
```

Рисунок 1 – Пример вывода программы

Таким образом, реализация циклического избыточного кодирования на языке C++ возможна и реализуется с помощью небольшого объема кода. Данный алгоритм неоптимизирован, но уже является применимым и помехоустойчивым к небольшим ошибкам.

Библиографический список

1. Williams R.N. Элементарное руководство по CRC алгоритмам обнаружения ошибок [Электронный ресурс]. URL: <http://web.archive.org/web/20130407000813/http://rdsn.ru/article/files/classes/SelfCheck/crcguide.pdf> (дата обращения 13.01.2017)
2. Простой расчет контрольной суммы [Электронный ресурс]. URL: <https://habrahabr.ru/post/278171/> (дата обращения 13.01.2017).
3. Циклический избыточный код [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/Циклический_избыточный_код (дата обращения 13.01.2017).