

```

1   Program crc_16;
2   {
    Задача. Написать табличную реализацию алгоритма CRC-16.

    Решение. Сергей Митрофанов,
            Центр НИТ "Северная Звезда",
            20.12.03, 17:49-20:00
            22.12.03, 11:58-20:00
            23.12.03, 10:00
    }

10  Uses Crt;

11  Const poly = $1021; { делитель-полином }

12  Var
13     TAB { таблица CRC для всех символов с кодами 0..255 }
14     : array [0..255] of word;

15     d, { длина текста }
16     crc, { контрольная сумма }
17     i { номер символа }
18     : word;

19     t, { текст }
20     t2 { выровненное сообщение }
21     : string;

22     symbol { символ }
23     : char;

24     first { старший байт аккумулятора контрольной суммы }
25     : byte;

26  Procedure print_16 (
27             a : longint
28             );
29  {
    печать числа в 16-ом виде
    }

32  const
33     dW = 16;
34     mask = $F;
35     mas_ts : array [0..15] of char
36             = ('0', '1', '2', '3', '4', '5', '6', '7',
37              '8', '9', 'A', 'B', 'C', 'D', 'E', 'F');
38  var
39     n, { шестнадцатеричная цифра в десятичном виде }

```

```

40     poz { на сколько бит произвести сдвиг }
41     : integer;

42 begin
43     Write (a, ' = $');

44     poz := dW - 4;
45     while poz > -1 do
46     begin
47         n := (a shr poz) and mask;
48         Write (mas_ts [n]);
49         poz := poz - 4;
50     end;
51 end;

52 Function poisk_W (
53     a { полином-делитель }
54     : longint
55     )
56     : integer;
57 {
58     вычисление степени полинома-делителя.
59     Поиск самого страшого бита, равного 1
60 }

61 const
62     dW = 16; { длина в битах двухбайтового целого числа }
63     mask = 1; { с чем будем сравнивать бит по and }

64 var
65     j, { число битовых единиц }
66     poz { номер бита, нумерация справа налево }
67     : integer;

68 begin
69     poz := dW - 1;
70     poisk_W := 0;
71     j := 0;
72     while poz > -1 do
73     begin
74         if (a shr poz) and mask = 1
75         then
76             begin
77                 Inc (j);
78                 if j = 1
79                 then
80                     begin
81                         poisk_W := poz;
82                         Exit;
83                     end;

```

```

84         end;
85         Dec (poz);
86     end;
87 end;

88 Procedure bits (
89         a { число }
90         : longint
91     );
92 {
93     DS, 22.12.03, 13:54
94     Найдем двоичный эквивалент знакового целого числа.
95 }

96 const
97     dW = 16; { длина в битах двухбайтового целого числа }
98     mask = 1; { с чем будем сравнивать бит по and }

99 var
100     poz { номер бита, нумерация справа налево }
101     : integer;

102 begin
103     print_16 (a);
104     Write (' = ');
105     poz := dW - 1;
106     while poz > -1 do
107     begin
108         if (a shr poz) and mask = 1
109         then
110             begin
111                 Write ('1');
112             end
113         else
114             Write ('0');
115         Dec (poz);
116     end;
117     WriteLn;
118 end;

119 Procedure make_tab;
120 {
121     DS, 23.12.03, 11:48
122     Создадим таблицу, в которой будут храниться CRC для всех
123     символов.
124 }

125 const
126     {

```

Существует стандартный 16-битный CRC-алгоритм CCITT CRC-16.  
Его параметры:  
имя: "CRC-16/СИТТ"  
длина делителя-полинома: 16 bits  
делитель-полином: \$1021  
инициализация значения CRC: \$FFFF

```
134   }  
      полином = $1021; { в десятичной системе это 4129 }  
  
135   var  
136     i, { номер элемента массива ТАВ, код символа }  
137     j { номер бита символа }  
138     : byte;  
  
139     crc { CRC для символа }  
140     : word;  
  
141   begin  
142     {  
        Коды всех символов: 0..255  
    }  
145   for i := 0 to 255 do  
146     begin  
147       crc := i shl 8;  
148       {  
          Просмотрим все биты текущего символа.  
          Посчитаем CRC для этого символа.  
        }  
152     for j := 1 to 8 do  
153       begin  
154         {  
            если старший бит crc равен 1  
          }  
157         if (crc and $8000) <> 0  
158           then  
159             begin  
160               crc := crc shl 1;  
161               crc := crc xor полином;  
162             end  
163           else  
164             crc := crc shl 1;  
165           end;  
166         {  
            поместим высчитанное CRC в таблицу CRC символов  
          }  
169         {  
            bits (crc);  
          }  
172         ТАВ [i] := crc;  
173       end;  
174   end; { end make_tab }
```

```

175     Begin
176         ClrScr;

177         { заполним таблицу CRC для всех символов 0..255 }
178         make_tab;

179         t := '12345';
180         symbol := Chr ($00);
181         {
            выровняем сообщение
        }
184         t2 := t + symbol + symbol;
185         WriteLn ('t2 = *', t2, '*');

186         d := Length (t2);
187         WriteLn ('d = ', d);

188         { инициализация перед началом расчета }
189         crc := $FFFF;
190         for i := 1 to d do
191             begin
192                 { найдем старший байт, first - это индекс рассчитанной TAB }
193                 first := (crc shr 8) and $FF;
194                 { дополним аккумулятор crc новым байтом из сообщения }
195                 crc := (crc shl 8) or Ord (t2 [i]);
196                 {
                    выполним операцию XOR
                    между аккумулятором и табличным значением
                }
200                 crc := crc xor TAB [first];
201             end;
202         {
            произведем побитное отрицание,
            получим контрольную сумму по алгоритму CRC-16
        }
206         crc := not crc;

207         Write ('crc = ');
208         bits (crc);

209         t2 := t + Chr ($5A) + Chr ($5D);

210         WriteLn ('t2 = *', t2, '*');

211         d := Length (t2);
212         WriteLn ('d = ', d);

213         crc := $FFFF;
214         for i := 1 to d do

```

```
215         begin
216             first := (crc shr 8) and $FF;
217             crc := (crc shl 8) or Ord (t2 [i]);
218             crc := crc xor TAB [first];
219         end;

220     crc := not crc;

221     Write ('crc = ');
222     bits (crc);

223     WriteLn;
224     WriteLn ('END. ');

225     ReadLn;
226     End.
```

Listing данной задачи опубликован в сети Internet по адресу  
<http://www.Best-Listing.ru/color-1-task-581.html>

Sergey Mitrofanov, 28.08.14, 19:24

E-mail: [infostar@mail.ru](mailto:infostar@mail.ru)

© <http://www.Best-Listing.ru/>, 2006–2014