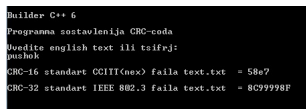


{comments on}

Разрабатываемая программа была написана в среде Builder C++ 6.

В данной программе рассчитывается CRC-16 standart CCITT с применением полинома $1021 = (x^{16} + x^{12} + x^5 + 1)$ и CRC-32 с применением полинома EDB88320UL = $(x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1)$ файла text.txt. Пользователю будет предложено ввести английское слово или цифры, затем сообщение пользователя будет передано в файл text.txt и после чего будет вывод на экран CRC-16 и CRC-32 изменённого файла text.txt. Далее информация, выведенная на экран будет продублирована в файл report.txt.

Пример работы программы



```
Builder C++ 6
Programma sostavlenija CRC-coda
Vvedite english text ili tsifry:
pushok
CRC-16 standart CCITT(mex) faila text.txt - 58e7
CRC-32 standart IEEE 802.3 faila text.txt - 8C99998F
```

Листинг программы:

```
/*Программа расчёта Crc32 для файлов.txt*/ #include <stdio.h>//необходимо для printf
#include <conio.h>//необходимо для getch #include <string.h>//необходимо для strlen
#define FNAME "text.txt"//макрос(константа) /*Описываем функцию расчёта Crc16 standart
CCITT с применением полинома  $1021 = x^{16} + x^{12} + x^5 + 1$ */ unsigned short
Crc16(unsigned char *buf, unsigned short len) { unsigned short crc = 0xFFFF;//переменная
16 бит = 2 байта unsigned char i; //переменная 8 бит = 1 байт while (len--)// проверка
условия продолжения { crc ^= *buf++ << 8; for (i = 0; i < 8; i++)//цикл перебора полинома
crc = crc & 0x8000 ? (crc << 1) ^ 0x1021 : crc << 1; } return crc;//конец функции расчёта
Crc16 } /*Описываем функцию расчёта Crc32 с применением полинома EDB88320UL =
 $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ */
unsigned long Crc32(unsigned char *buf, unsigned long len) { //инициализируем таблицу
расчёта Crc32 unsigned long crc_table[256];//массив 32 бита = 4 байтам unsigned long
crc;//переменная 32 бита = 4 байтам for (int i = 0; i < 256; i++)//инициализируем цикл
массива { crc = i; for (int j = 0; j < 8; j++)//цикл перебора полинома crc = crc & 1 ? (crc >> 1)
^ 0xEDB88320UL : crc >> 1; crc_table[i] = crc;}; crc = 0xFFFFFFFFFUL; while (len--)//
проверка условия продолжения crc = crc_table[(crc ^ *buf++) & 0xFF] ^ (crc >> 8); return
crc ^ 0xFFFFFFFFFUL; //конец функции расчёта Crc32 }; /*Создаём файл text.txt для
записи и обработки*/ int main() { char c[50];//массив для вводимого текста FILE *f =
fopen("text.txt", "w");//Открыли файл text.txt для записи FILE *fp = fopen("report.txt",
"w");//Открыли файл report.txt для записи if ((f = fopen("text.txt", "w"))==NULL)//Если файл
```

```
невозможно открыть { printf("nError = Nevozmozchno otcrit' failn");//то выдаём сообщение  
getch();//задерживаем программу return 0;//выходим из программы } else // { //выдаём  
следующие сообщения printf("nBuilder C++ 6n"); printf("nProgramma sostavlenija  
CRC-codan"); printf("nVvedite english text ili tsifry:n"); scanf("%s",&c);//вводим текст fprintf(f,  
"%s", c);//передаём введённый текст в файл text.txt fclose(f); //Файл text.txt закрыли } if  
((f = fopen("text.txt", "rt"))==NULL)//Если файл невозможно открыть { printf("nError =  
Nevozmozchno otcrit' failn");//то выдаём сообщение getch();//задерживаем программу  
return 0;//выходим из программы } else//иначе unsigned long Crc32(unsigned char *buf,  
unsigned long len); char buf[4096]; int len; fgets(buf, sizeof(buf), f); len =  
strlen(buf);//Выводим на консоль имя файла и его CRC-16 и CRC-32 printf("nCRC-16  
standart CCITT(nex) faila %s = %xn",FNAME,Crc16(buf,len)); printf("nCRC-32 standart IEEE  
802.3 faila %s = %pn",FNAME,Crc32(buf,len)); //передаём выведенный текст в файл  
report.txt fprintf(fp, "REPORT File %s", FNAME); fprintf(fp, "nCRC-16 standart CCITT(nex) =  
%xn",Crc16(buf,len)); fprintf(fp, "CRC-32 standart IEEE 802.3 = %pn",Crc32(buf,len));  
fclose(f);//Файл text.txt закрыли fclose(fp);//Файл report.txt закрыли getch();//задержка  
программы return 0; //конец программы }
```

```
(function(w, d, n) { w[n] = w[n] || []; w[n].push({ section_id: 263974, place: "advertur_263974",  
width: 300, height: 250 }); })(window, document, "advertur_sections");
```