

1. Treść zadania. Napisać program dopisujący do pliku CRC32 oraz sprawdzający integralność danych w pliku z dopisanym CRC.

2. Realizacja

2.1 Metoda realizacji

Program oblicza wartość sumy kontrolnej którą wylicza algorytm CRC z wykorzystaniem funkcji XOR dla bitów zapisanych w pliku tekstowym. Na początku program wczytuje od użytkownika nazwę pliku który chcemy zaszyfrować oraz wielomian CRC (wybrałem 0x82608EDB). Po wyświetleniu wszystkich kroków operacji xor, program wysyła do wybranego przez nas pliku. Program zawiera też metodę kontrolną w której sprawdzamy czy zakodowana wiadomość dzieli się bez reszty z wielomianem CRC.

Treść zadania jest zrealizowana w języku C++ z użyciem operacji na plikach, operacjach na string oraz pętli.

2.2 Ograniczenia metody:

2.2.1 Plik wejściowy musi być liczbą binarną, nie mam zaimplementowanej w kodzie metody zamiany tekstu na ascii a potem na bity. Jeśli chcemy zaszyfrować tekst musimy ją zamienić przed uruchomieniem programu.

2.2.2 Chciałem pokazać jak działa crc w każdym kroku, jednak dla większych kodów nie ma to sensu, dlatego można usunąć printf w pętli XOR

2.3 Realizacja:

OBLICZANIE CRC32

Main()

{

2.3.1 Wczytuje plik

2.3.2 Dane z pliku zapisuje w stringu

2.3.3 Nadaje 32 bitowy wielomian CRC

2.3.4 Dodaje do dlugosci stringa wczytanego 32 zera na koncu

XOR(wczytany string, wybrany wielomian CRC)

{

2.3.5 petla o dlugosci (przemieszczajaca się do 32 miejsca)

2.3.6 xor

}

2.3.7 zapisuje zakodowany tekst do nowego pliku

}

SPRAWDZANIE CRC32

2.3.8 wczytuje plik z kodem źródłowym +CRC

2.3.9 przechodzę do operacji XOR z wielomianem startowym

2.3.10 jeśli operacja XOR dzieli się bez reszty nie ma błędu w komunikacji.

Kod źródłowy:

```
#include<iostream>
```

```
#include <fstream>
```

```
using namespace std;
```

```
string xorfun( string encoded , string crc)
```

```
    //Bitowa operacja XOR
```

```
{
```

```
int crclen = crc.length();
```

```
for ( int i = 0 ; i <= (encoded.length() - crclen) ; )
```

```
// petla
```

```
wykonujaca operacje xor
```

```

{
    // " 0 xor 0 = 0"   " 1 xor 1 = 0 "

    for( int j=0 ; j < crcLen ; j++) // " 0 xor 1
    = 1 "   " 1 xor 0 = 1"

    {
        encoded[i+j] = encoded[i+j] == crc[j] ? '0' : '1' ; //jesli bit z pliku i
        bit wielomianu jest taki sam to zastap je zerem

        cout<<encoded<<endl;
    }
    for( ; i< encoded.length() && encoded[i] != '1' ; i++) ;

}

return encoded;
}

int main()
{
    string data , crc , encoded = "";
    string myText;
    cout<<endl<<"-----TWORZENIE CRC -----"<<endl;
    cout<<"Podaj nazwe pliku: "<<endl;
    cin>>data;

    ifstream MyReadFile(data);

    while (getline (MyReadFile, myText)) {
        cout << myText<<endl; }

    MyReadFile.close();
}

```

```
crc="10000010011000001000111011011011";
cout<<"Wielomian generujacy: "<<crc<<endl;

encoded += myText;

int datalen = myText.length();
int crclen = crc.length();

for(int i=1 ; i <= (crclen - 1) ; i++)
    encoded += '0';

encoded = xorfun(encoded , crc);

cout<<"Cykliczny kod nadmiarowy to: ";
cout<<encoded.substr(encoded.length() - (crclen-1) )<<endl<<endl;
cout<<myText + encoded.substr(encoded.length() - crclen + 1)<<endl;

    string data2;
    cout<<"Podaj nazwe pliku do zapisania: "<<endl;
    cin>>data2;

    ofstream MyFile(data2);

    MyFile <<(myText + encoded.substr(encoded.length() - crclen + 1));
    MyFile.close();
```

```

cout<<endl<<"-----SPRWDZENIE-----"<<endl;

    cout<<"Podaj nazwe pliku z zaszyfrowana wiadomoscia: "<<endl;
    string msg, data1;
    cin>>data1;
    ifstream MyReadFile1(data1);
    getline (MyReadFile1, msg);
    MyReadFile1.close();
    cout<<"Zakodowana wiadomosc to: "<<endl<<msg<<endl<<endl;

    msg = xorfun( msg , crc);

    for( char i : msg.substr(msg.length() - crclen + 1))
    if( i != '0' )
    {
        cout<<"Blad w komunikacji "<<endl;
        return 0;
    }

    cout<<"Brak bledu !"<<endl;
    return 0;
}

```

2.3 Uruchomiony program:

2.3.1 Plik z liczba binarną:

a) z każdą kolejną operacją

[illegible]

b) bez wypisywania operacji

[illegible]

2.3.5

Podaj nazwe pliku do zapisania:
kod_binarny+CRC.txt

-----SPRAWDZENIE-----

2.3.6 Wczytujemy plik w którym zapisaliśmy kod binarny z obliczonym CRC

[illegible]

2.3.7 Poprawny kod :

[illegible]

2.3.7 Podmieniony kod (bez wypisywania wszystkich operacji xor):

WybierzC:\Users\Wojtek\Downloads\zaddaad\bin\Debug\zaddaad.exe

```
-----TWORZENIE CRC -----
```

Podaj nazwe pliku:

`kod_binarny.txt`

[illegible]

Wielomian generujący: $10000010011000001000111011011011$

Cykliczny kod nadmiarowy to: 110111110100101001010010100010

[illegible]

Podaj nazwe pliku do zapisania:

kod_biarny+CRC.txt

-----SPRWDZENIE-----

Podaj nazwe pliku z zaszyfrowana wiadomoscia:

kod_biarny+CRC.txt

Zakodowana wiadomosc to:

[illegible]

Błąd w komunikacji

```
Process returned 0 (0x0)   execution time : 66.875 s
```

Press any key to continue.