***Wojskowa Akademia Techniczna***

***im. Jarosława Dąbrowskiego***

Laboratorium z przedmiotu:

Interfejsy komputerów cyfrowych

Sprawozdanie nr 2:

**Obliczanie oraz sprawdzanie CRC dla pliku**

Prowadząca:

mgr inż. Małgorzata Pisula

**Wykonał:** Radosław Relidzyński

**Grupa:** WCY20IY4S1

**Data laboratoriów**: 10.06.2022 r.

Spis treści

[Kompletny program 2](#_Toc105745635)

[Przykład działania (plik: „smalltest.txt”) 5](#_Toc105745636)

[Funkcja obliczająca CRC 7](#_Toc105745637)

[Analiza kalkulowania CRC 7](#_Toc105745638)

[Analiza sprawdzania CRC 8](#_Toc105745639)

# Kompletny program

#include<iostream>

#include<string>

#include<sstream>

#include<fstream>

#include<bits/stdc++.h>

using namespace std;

int m = 0xEDB88320;

unsigned int crc\_tab[256];

int ConvertToDecimal(int d, string s){

    if(d==1){

        if((s[0] - 55) < 10){

            return int(s[0] - 48);

        }

        else{

            return int(s[0] - 55);

        }

    }

    else{

        if((s[d-1] - 55) < 10){

            return 16 \* ConvertToDecimal(d-1, s) + int(s[d-1] - 48);

        }

        else{

            return 16 \* ConvertToDecimal(d-1, s) + int(s[d-1] - 55);

        }

    }

}

unsigned int CalculateCRC(string input){

    unsigned int crc = 0xFFFFFFFF, cond;

    for(unsigned char x : input){

        crc=(crc>>8) ^ crc\_tab[(crc ^ x) & 0xFF];

    }

    return ~crc;

}

int main(){

    unsigned int tmp;

    // crc\_tab has all values needs for crc calculation equation

    for(int i=0; i<256; i++){

        tmp=i;

        for(int j=0; j<8; j++){

            tmp=(tmp >> 1) ^ (tmp%2 \* m);

        }

        crc\_tab[i]=tmp;

    }

    // Open file

    char input\_file\_name[] = "smalltest.txt";

    FILE \*input\_file = fopen(input\_file\_name, "rb");

    if (!input\_file) {

        cout << "Can't open file";

        exit(0);

    }

    // Read text from file

    unsigned char buffer[1];

    string input\_text = "";

    while (fread(buffer, sizeof(unsigned char), 1, input\_file)){

        input\_text += buffer[0];

    }

    fclose(input\_file);

    // Choose operation

    cout << "1. Calculate CRC\n2. Check CRC\n";

    int c=-1;

    cin >> c;

    // 1. Calculate CRC

    if(c==1){

        cout << "Calculating CRC\n";

        unsigned int result = CalculateCRC(input\_text);

        cout << "Calculated CRC: 0x" << hex << result;

        ofstream input\_file;

        input\_file.open(input\_file\_name, ios::out | ios::binary);

        if (!input\_file) {

            cout << "Can't open file";

            exit(0);

        }

        input\_file.seekp(0, ios\_base::end);

        input\_file<<input\_text;

        string hex\_crc;

        unsigned int r = result;

        while(r > 0){

            if(r % 16 >= 10) hex\_crc = char(r % 16 + 55) + hex\_crc;

            else hex\_crc = to\_string(r % 16) + hex\_crc;

            r /= 16;

        }

        string tmp, add, crc\_s;

        for(int i=0; i<hex\_crc.length(); i+=2){

            tmp+=hex\_crc[i];

            tmp+=hex\_crc[i+1];

            add = char(ConvertToDecimal(2, tmp));

            crc\_s += add;

            input\_file<<char(add[0]);

            tmp.erase();

        }

        cout << "\nAdded " << crc\_s << " to file\n";

    }

    // 2. Check CRC

    if(c==2){

        string crc = input\_text.substr(input\_text.length()-4, 4);

        string crc\_hex\_test;

        cout << "Calculating CRC\n";

        unsigned int result = CalculateCRC(input\_text.substr(0, input\_text.length()-4));

        cout << "Calculated CRC:\t0x" << hex << result;

        // Reading CRC

        for(int i=0; i<crc.length(); i++){

            string hex\_crc;

            // Convert to hex

            unsigned char r = crc[i];

            while(r > 0) {

                if (r % 16 >= 10) hex\_crc = char(r % 16 + 87) + hex\_crc;

                else hex\_crc = to\_string(r % 16) + hex\_crc;

                r /= 16;

            }

            crc\_hex\_test += hex\_crc.length() == 2 ? hex\_crc : char(48) + hex\_crc;

        }

        cout<<"\nReaded CRC:\t0x"<< hex << crc\_hex\_test <<endl;

        ostringstream r;

        r << hex << result;

        string res = r.str();

        ostringstream xx;

        xx << hex << crc\_hex\_test;

        string x = xx.str();

        //if(result == ConvertToDecimal(crc\_hex\_test.length(), crc\_hex\_test)){

        if(res == x){

            cout<<"CRC is correct";

        }

        else{

            cout<<"File is corrupted, CRC incorrect";

        }

    }

    return 0;

}

# Przykład działania (plik: „smalltest.txt”)

Zawartość pliku:

Obraz zawierający tekst, stół

Opis wygenerowany automatycznie

Konsola - kalkulowanie:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Zawartość pliku z dopisanym CRC:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Konsola – sprawdzanie:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Zmiana bajtu w pliku:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Konsola – sprawdzanie po zmianie pliku:

Obraz zawierający tekst

Opis wygenerowany automatycznie

# Funkcja obliczająca CRC

1. crc = 0xFFFFFFFF, obliczenia na tej zmiennej będą zmieniać jej wartość do momentu obliczenia crc.
2. Dla każdego odczytanego znaku:
   1. Wykonaj przesunięcie bitowe o 1 bajt.
   2. Pobierz wartość tablicy na podstawie ostatniego bajtu zmiennej.
   3. Wykonaj operację xor między tymi wartościami.
   4. Wpisz ją do zmiennej crc.
3. Zwróć negację zmiennej crc.

unsigned int CalculateCRC(string input){

    unsigned int crc = 0xFFFFFFFF, cond;

    for(unsigned char x : input){

        crc=(crc>>8) ^ crc\_tab[(crc ^ x) & 0xFF];

    }

    return ~crc;

}

# Analiza kalkulowania CRC

1. Do zmiennej result przypisuję wartość funkcji CalculateCRC() na podstawie pobranego tekstu z pliku.
2. Otwieram analizowany plik w trybie pisania, idę na jego koniec linii
3. Konwertuję crc do typu string:
   1. Jeśli wartość jest większa lub równa 10, to wpisuję odpowiednią literę.
   2. Jeśli wartość jest mniejsza od 10, wpisuję jej wartość.
4. Znak po znaku wpisuję kolejne bajty do pliku.

    // 1. Calculate CRC

    if(c==1){

        cout << "Calculating CRC\n";

        unsigned int result = CalculateCRC(input\_text);

        cout << "Calculated CRC: 0x" << hex << result;

        ofstream input\_file;

        input\_file.open(input\_file\_name, ios::out | ios::binary);

        if (!input\_file) {

            cout << "Can't open file";

            exit(0);

        }

        input\_file.seekp(0, ios\_base::end);

        input\_file<<input\_text;

        string hex\_crc;

        unsigned int r = result;

        while(r > 0){

            if(r % 16 >= 10) hex\_crc = char(r % 16 + 55) + hex\_crc;

            else hex\_crc = to\_string(r % 16) + hex\_crc;

            r /= 16;

        }

        string tmp, add, crc\_s;

        for(int i=0; i<hex\_crc.length(); i+=2){

            tmp+=hex\_crc[i];

            tmp+=hex\_crc[i+1];

            add = char(ConvertToDecimal(2, tmp));

            crc\_s += add;

            input\_file<<char(add[0]);

            tmp.erase();

        }

        cout << "\nAdded " << crc\_s << " to file\n";

    }

# Analiza sprawdzania CRC

1. Pobieram z pliku jego crc (ostatnie 4 bajty).
2. Na podstawie samego tekstu obliczam crc z funkcji.
3. Konwertuję pobrane crc do postaci hexadecymalnej.
4. Porównuję pobrane crc z obliczomym crc:
   1. Jeśli jest takie samo, wszystko jest w porządku.
   2. Jeśli jest różne, znaczy to, że plik został naruszony.

    // 2. Check CRC

    if(c==2){

        string crc = input\_text.substr(input\_text.length()-4, 4);

        string crc\_hex\_test;

        cout << "Calculating CRC\n";

        unsigned int result = CalculateCRC(input\_text.substr(0, input\_text.length()-4));

        cout << "Calculated CRC:\t0x" << hex << result;

        // Reading CRC

        for(int i=0; i<crc.length(); i++){

            string hex\_crc;

            // Convert to hex

            unsigned char r = crc[i];

            while(r > 0) {

                if (r % 16 >= 10) hex\_crc = char(r % 16 + 87) + hex\_crc;

                else hex\_crc = to\_string(r % 16) + hex\_crc;

                r /= 16;

            }

            crc\_hex\_test += hex\_crc.length() == 2 ? hex\_crc : char(48) + hex\_crc;

        }

        cout<<"\nReaded CRC:\t0x"<< hex << crc\_hex\_test <<endl;

        ostringstream r;

        r << hex << result;

        string res = r.str();

        ostringstream xx;

        xx << hex << crc\_hex\_test;

        string x = xx.str();

        //if(result == ConvertToDecimal(crc\_hex\_test.length(), crc\_hex\_test)){

        if(res == x){

            cout<<"CRC is correct";

        }

        else{

            cout<<"File is corrupted, CRC incorrect";

        }

    }