*Politechnika Warszawska*

SPUP

Wykonał: Mateusz Jaworski 33ip

289702

**Zadanie 2**

**Kod programu:**

Cały program dostępny na githubie: <https://github.com/jaworskimateusz/university-projects/tree/master>

Uruchomienie:

1. Klonujemy repozytorium: git clone https://github.com/jaworskimateusz/university-projects.git
2. Importujemy projekt i odpalamy plik z rozszerzeniem .pro

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <string>

#include <iostream>

#include <cstdlib>

#include <math.h>

#include <time.h>

#include <QElapsedTimer>

#include <QByteArray>

#include <QString>

*using* *namespace* std;

*namespace* **Ui** {

*class* **MainWindow**;

}

*class* **MainWindow** : *public* QMainWindow {

Q\_OBJECT

*static* *const* uint32\_t MAX\_LENGTH = 256;

QString byteCode;

int numberOfReps;

int registry;

*public*:

*explicit* **MainWindow**(QWidget \*parent = 0);

~***MainWindow***();

void **calculate**();

std::string **toHex**(int decimal);

*private* slots:

void **onCalculate**();

*private*:

Ui::MainWindow \*ui;

};

#endif *//* *MAINWINDOW\_H*

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <iostream>

#include <QRegExp>

#include <QtCore/QtEndian>

#include <QShortcut>

#include <QDebug>

#include <QRegExpValidator>

MainWindow::**MainWindow**(QWidget \*parent) :

QMainWindow(parent),

ui(*new* Ui::MainWindow) {

ui->setupUi(*this*);

QRegExp HEXregExp("[A-Fa-f0-9]{0,}");

QRegExp NUMregExp("[0-9]{0,}");

ui->byteCode->setValidator(*new* QRegExpValidator(HEXregExp, *this*));

ui->numberOfReps->setValidator(*new* QRegExpValidator(NUMregExp, *this*));

connect(MainWindow::findChild<QPushButton \*>("calculate"), SIGNAL(released()),*this*, SLOT(onCalculate()));

}

MainWindow::~***MainWindow***() {

*delete* ui;

}

void MainWindow::**onCalculate**() {

ui->time->setText(QString::number(0));

ui->crc16->setText(QString::number(0));

byteCode = ui->byteCode->text();

numberOfReps = ui->numberOfReps->text().toInt();

QElapsedTimer timer;

*if*(byteCode.length() % 2 == 0) {

timer.start();

*for* (int i = 0; i < numberOfReps; i++) {

registry = 0xFFFF;

*for* (int j = 0; j < byteCode.length(); j += 2) {

registry ^= stoul(byteCode.mid(j,2).toStdString(), NULL, 16);

*for* (int k = 0; k < 8; k++) {

*if* (registry & 0x0001) {

registry >>= 1;

registry ^= 0xA001;

} *else* {

registry >>= 1;

}

}

}

}

ui->crc16->setText(QString::fromStdString(toHex(registry & 0xFF)) + QString::fromStdString(toHex(registry >> 8)));

ui->time->setText(QString::number(timer.elapsed()));

}

}

std::string MainWindow::**toHex**(int decimal) {

int residual;

std::string hex = "";

*while* (decimal > 0) {

residual = decimal % 16;

*if* (residual > 9) {

*switch* (residual) {

*case* 10: hex = "A" + hex; *break*;

*case* 11: hex = "B" + hex; *break*;

*case* 12: hex = "C" + hex; *break*;

*case* 13: hex = "D" + hex; *break*;

*case* 14: hex = "E" + hex; *break*;

*case* 15: hex = "F" + hex; *break*;

}

} *else* {

hex = char(residual + 48) + hex;

}

decimal = decimal/16;

}

*return* hex;

}

**main.cpp**

#include "mainwindow.h"

#include <QApplication>

int **main**(int argc, char \*argv[])

{

QApplication a(*argc*, argv);

MainWindow w;

w.show();

*return* a.exec();

}

**Krótki opis softu:**

Program napisany został w języku C++ w środowisku QT Creator. Layout wykonanu w bibliotece dostępnej w tym środowisku.

Byte Code wporwadzamy napis w notacji heksadecymalnej.

Number of Reps ilość powtórzeń algorytmu.

CRC16 wynik naszych obliczeń.

Time czas trwania w ms.

Calculate przycisk pozwalający na wykonanie operacji.

A screenshot of a cell phone

Description automatically generated

Zrzut ekranu 1. Przykładowy rezultat dla mniejszej ilości powtórzeń

**Wyznaczenie CRC:**

Dla sekwencji z zadania z ilością powtórzeń 1000000000 uzyskałem wynik 7f67, czas wykonania był dość długi i wynosił 482456 ms (ok.8 min)

**Parametry sprzętu:**

Laptop Apple MacBook Pro:

- procesor – Intel Core i7 3,3 GHz

- 16 GB pamięci RAM