ГУАП

КАФЕДРА № 44

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | К. А. Белов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4 |
| ПОЛУЧЕНИЕ ИСПОЛНЯЕМОГО ФАЙЛА ДЛЯ ПЛАТФОРМЫ «ЭЛЬБРУС» |
| по курсу: ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ ЭЛЬБРУС |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |
| --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4941 |  | Н. С. Горбунов |
|  |  | подпись, дата | инициалы, фамилия |

Санкт-Петербург

2023

# Цель работы

Целью данной лабораторной работы является: ознакомление с основными свойствами и поддерживаемыми событиями стандартных компонентов, применяемых для создания интерфейсов при разработке графических приложений, с использованием кроссплатформенных библиотек доступных в ОС «Эльбрус-Линукс».

# Исходные данные варианта

**Вариант задания 4:**

Умножение матрицы А размером И х Ф на матрицу В размером Ф х О;начальные значения вещественны и задаются по формуле:

|  |  |
| --- | --- |
| Используемая библиотека | № по списку группы |
| QT | 7 |

# Задания на выполнение

1. Для задания из ЛР №3, либо по варианту составить макет пользовательского интерфейса;
2. Выполнить описание поставленной задачи и выбранного метода (библиотеки) решения;
3. Разработать исполняемый модуль;
4. Разработать тестовые примеры
5. Подготовить отчёт по ЛР.

# Процесс выполнения работы

* 1. **Описание средств разработки**

Qt позволяет запускать написанное с его помощью программное обеспечение в большинстве современных операционных систем путём простой компиляции программы для каждой системы без изменения исходного кода. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Является полностью объектно-ориентированным, расширяемым и поддерживающим технику компонентного программирования.

# Описание поставленной задачи

За основу для приложения была взята программа из лабораторной работы № 3 – программа, выполняющая умножение матриц. Для того, чтобы перенести консольное приложение на платформу QT необходимо воспользоваться средой разработкой QT designer для составления графического интерфейса программы.

# Результат разработки

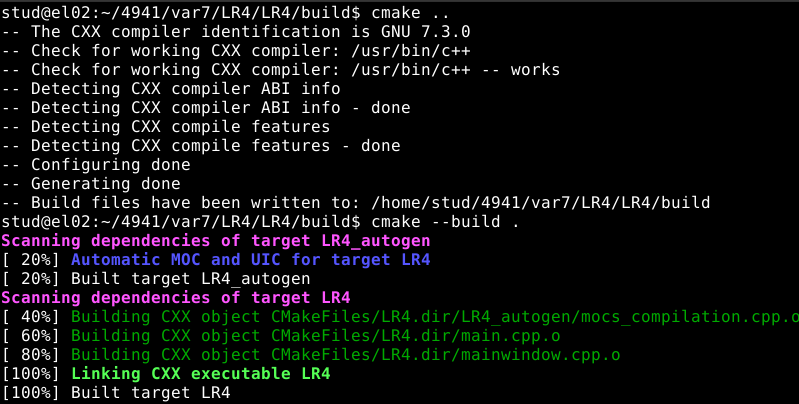


Рисунок 1 – Сборка на ОС ЭльбрусЛинукс

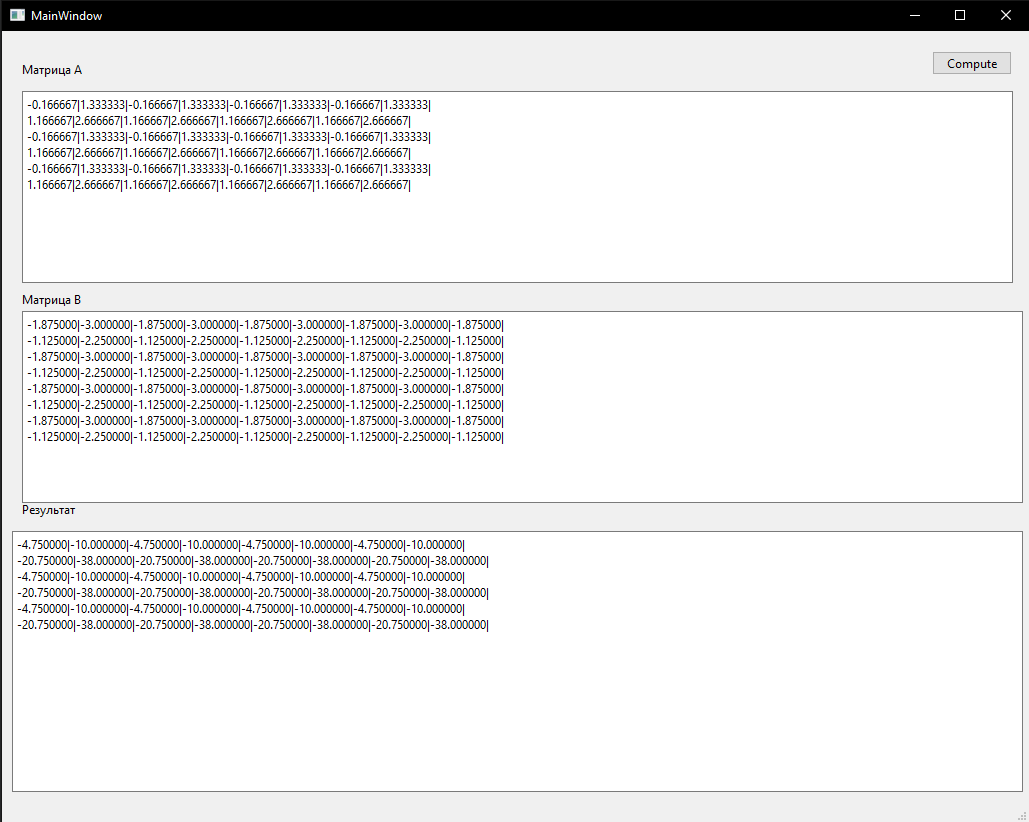


Рисунок 2 – Интерфейс программы

# Исходный код программы

Файл main.cpp:

#include "mainwindow.h"

#include <QApplication>

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

{

QApplication a(argc, argv);

MainWindow w;

w.show();

return a.exec();

}

Файл mainwidnow.cpp:

#include "mainwindow.h"

#include "./ui\_mainwindow.h"

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::on\_ComputeButton\_clicked()

{

// Ф-8, И-6, О-9

int f = 8, n = 6, o = 9, i, j, k;

float a[6][8], b[8][9], mult[6][8];

for (i = 0; i < n; i++)

{

for (j = 0; j < f; j++)

{

a[i][j] = ((float)(f \* ((i % 2) + 1) + o \* ((j % 2) - 1))) / (float)n;

}

}

for (i = 0; i < f; i++)

{

for (j = 0; j < o; j++)

{

b[i][j] = ((float)(n \* ((i % 2) - 1) - o \* ((j % 2) + 1))) / (float)f;

}

}

//Matrix A

QString astr = "";

for (i = 0; i < n; i++)

{

for (j = 0; j < f; j++)

{

astr+= QString::fromStdString( std::to\_string(a[i][j])) + "|";//cout << a[i][j] << " ";

}

astr+= "\n";

}

ui->aMatrixText->setPlainText(astr);

//Matrix B

QString bstr = "";

for (i = 0; i < f; i++)

{

for (j = 0; j < o; j++)

{

bstr+= QString::fromStdString( std::to\_string(b[i][j])) + "|";

}

bstr+= "\n";

}

ui->bMatrixText->setPlainText(bstr);

// Multiplying matrix a and b and storing in array mult.

for (i = 0; i < n; i++)

{

for (j = 0; j < o; j++)

{

mult[i][j] = 0;

for (k = 0; k < f; k++)

{

mult[i][j] += a[i][k] \* b[k][j];

}

}

} // Displaying the multiplication of two matrix.

//Output Matrix:

QString resultstr = "";

for (i = 0; i < n; ++i)

{

for (j = 0; j < f; ++j)

{

resultstr+= QString::fromStdString( std::to\_string(mult[i][j])) + "|";

}

resultstr+= "\n";

}

ui->resultText->setPlainText(resultstr);

}