Федеральное государственное бюджетное образовательное учреждение высшего образования «Национальный исследовательский университет «МЭИ»

Институт информационных и вычислительных технологий

Кафедра Управления и интеллектуальных технологий

**Отчёт по лабораторной работе № 4**

# По курсу «Разработка ПО систем управления»

# 1 часть - «Библиотеки»

# 2 часть – «Низкоуровневые операции»

# Выполнил студент группы А-02-19

# Козлов И. А

# Проверили

# Мохов А. С.

# Козлюк Д. А

2020

1.

Заменяем cin на in в input\_numbers(istream& in, size\_t count)

Доводим программу до прежней работы, делаем коммит.

2.

После выполнения пункта 3.1.2 main приняла такой вид

// Ввод данных

const auto input = read\_input(cin, true);

// Вывод данных

const auto bins = make\_histogram(input);

// show\_histogram\_text(bins, numbers, number\_count, bin\_count);

show\_histogram\_svg(bins, input);

Добились ввода с помощью структуры.

3. Подключение curl

Добавляя первой строкой в main

curl\_global\_init(CURL\_GLOBAL\_ALL);

Возникают ошибки:

error: 'CURL\_GLOBAL\_ALL' was not declared in this scope

error: 'curl\_global\_init' was not declared in this scope

Добавляем файл #include <curl/curl.h>

Однако всё равно выдаёт ошибку: файл не найден

Добавив через project->built options пути поиска include, ошибка меняется на

undefined reference to `\_imp\_\_curl\_global\_init'|

Возвращаемся к Search directories -> Linker. Добавляем каталог curl/lib

Затем libcurl.dll.a в настройке линкера.

Программа собирается, однако при запуске выдаёт ошибку.

4.

$ ./lab01.exe -x --y /z w

argv[0]=C:\Users\▒▒▒▒▒\Desktop\lab03\Lab03\bin\Debug\lab01.exe12142 argv[1]=-x12142 argv[2]=--y12142 argv[3]=Z:/12142 argv[4]=w12142

Enter number\_count:

$ ./lab01.exe param "with spaces"

argc[0]=C:\Users\▒▒▒▒▒\Desktop\lab03\Lab03\bin\Debug\lab01.exe12142argc[1]=param12142argc[2]=with spaces12142Enter number\_count:

$ ./lab01.exe param <marks.txt 2>NUL

<?xml version='1.0' encoding='UTF-8'?>

<svg width='500' height='300' viewBox='0 0 500 300' xmlns='http://www.w3.org/2000/svg'>

<text x='20' y='20'>2</text>

<rect x='50' y='0' width='20' height='30' stroke='black' fill='green' />

<text x='20' y='50'>5</text>

<rect x='50' y='30' width='50' height='30' stroke='black' fill='red' />

<text x='20' y='80'>3</text>

<rect x='50' y='60' width='30' height='30' stroke='black' fill='green' />

<text x='50' y='120'>Windows v6.2 (build 9200) Computer name: DESKTOP-COR9GBJ</text></svg>

5. Загрузка файла по сети.

Весь код выполняется в только в случае argc > 1

CURL \*curl = curl\_easy\_init();

if(curl)

{

CURLcode res;

curl\_easy\_setopt(curl, CURLOPT\_URL, address.c\_str());

curl\_easy\_setopt(curl, CURLOPT\_WRITEFUNCTION, write\_data);

curl\_easy\_setopt(curl, CURLOPT\_WRITEDATA, &buffer);

res = curl\_easy\_perform(curl);

if(res!=0)

{

string error = curl\_easy\_strerror(res);

cout << "ERROR: " << error;

exit(1);

}

curl\_easy\_cleanup(curl);

}

В итоге функция main принимает такой вид:

int

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

if (argc > 1) {

*// работа с cURL*

return 0;

}

*// прежний код построения гистограммы*

}

curl\_easy\_perform() – финкция обработки ошибок, переменная res возвращает номер ошибки.

6. Работа с буфером

Для сохранения данных создадим функцию download(), в переменную типа stringstream можно записывать данные, то етсь сохранять в буфер.

С помощью callback функции writedata записываем данные в буфер прямо во время считывания данных из сети.

2 часть lab04

В этой части требуется вывести под гистограммой версию компьютера, сборку и его имя.

Так же весь вывод информации в этой части лабораторной будет совершен с помощью printf();



В этой части программы так же будем использовать буфер.

С помощью функции GetVersion получаем версию компьютера, но информация о версии хранится в битах этого числа, нужно расшифровать полученные данные.

2.

Из документации узнаем, в каких битах находится мажорный и минорный номер программы: номер версии ОС в младшем слове и информацию о платформе в старшем слов.

С помощью так называемых «масок» оставляем нужные нам байты и с помощью побитовых операций достаем нужные байты и печатаем числа уже в десятичном виде, получаем версию.

младшее слово содержит номер версии ОС: младший байт — мажорный номер, старший байт — минорный.

3.

Чтобы получить номер сборки, нужно взять 8 бит самого старшего слова, в настоящее время сборки кодируются 16 битами самого старшего слова. Если старший бит нулевой, можно получить число, которое составляют 15 следующих бит, как просто два старших байта.

С помощью маски оставляем 15 нужных бит и печатаем номер сборки.

4.

С именем компьютера все проще, нужна лишь функция GetComputerNameA()

DWORD size = 256;

char ComputerName[size];

GetComputerNameA(ComputerName, &size);

Размер массива узнаем из документации.

5.

С помощью svg функций выводим все полученные данные.

Проверяем работу программы и коммитим изменения.