**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет информационных технологий Кафедра параллельных вычислений**

**ОТЧЕТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

## «Низкоуровневая работа с периферийными устройствами» Студента 2 курса, 21202 группы

**Куращенко Льва Владиславовича**

## Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

## Перепелкин В.А.

Новосибирск 2022

# ЦЕЛЬ

* *Ознакомиться с началами низкоуровневого программирования периферийных устройств на примере получения информации о доступных USB-устройствах с помощью библиотеки libusb;*

# ЗАДАНИЕ

1. *Реализовать программу, получающую список всех подключенных к машине USB устройств с использованием libusb. Для каждого найденного устройства напечатать его класс, идентификатор производителя и идентификатор изделия.*
2. *Изучить состав и характеристики обнаруженных с помощью реализованной программ USB устройств.*
3. *Дополнить программу функцией печати серийного номера USB устройства. Для написания функции рекомендуется использовать функции libusb\_open, libusb\_close, libusb\_get\_string\_descriptor\_ascii для печати поля iSerialNumber дескриптора устройства.*
4. *Составить отчет по лабораторной работе. Отчет должен содержать следующие пункты:*
   * *Титульный лист;*
   * *Цель лабораторной работы;*
   * *Полный компилируемый листинг реализованной программы и команды для ее компиляции;*
   * *Описание обнаруженных USB-устройств;*
   * *Вывод по результатам лабораторной работы.*

# Листинг программы

#include "libusb-1.0/libusb.h"

#include <stdio.h>

#include <cstring>

/\*

\* How to execute:

\* >g++ Usb.cpp -o usb -lusb-1.0

\* >./usb

\*

\*

\*

\*/

void PrintFields(libusb\_device \*device) {

struct libusb\_device\_descriptor descriptor;

libusb\_get\_device\_descriptor(device,&descriptor);

struct libusb\_device\_handle \*handle;

libusb\_open(device,&handle);

unsigned char iSerialNumber[256];

if (handle && descriptor.iSerialNumber) {

libusb\_get\_string\_descriptor\_ascii(handle, descriptor.iSerialNumber, iSerialNumber, sizeof(iSerialNumber));

}

else

strncpy(reinterpret\_cast<char \*>(iSerialNumber), "null", sizeof(iSerialNumber));

libusb\_close(handle);

printf("%-20.2x 0x%-18.4x 0x%-18.4x %-20s\n", (int) descriptor.bDeviceClass, descriptor.idVendor, descriptor.idProduct, iSerialNumber);

}

int main() {

libusb\_device \*\*device;

libusb\_context \*context;

libusb\_init(&context);

ssize\_t devices\_amount = libusb\_get\_device\_list(context, &device);

printf("Devices found: %zd\n\n", devices\_amount);

printf("%-20s %-20s %-20s %-20s\n","Device class","Vendor ID","Product ID","Serial number\n");

for (int i = 0; i < devices\_amount; i++) {

PrintFields(device[i]);

}

libusb\_free\_device\_list(device, 1);

libusb\_exit(context);

return 0;

}

# Вывод программы

# 

# ЗАКЛЮЧЕНИЕ

*В ходе выполнения лабораторной работы:*

* *Ознакомился с началами низкоуровневого программирования периферийных устройств на примере получения информации о доступных USB-устройствах с помощью библиотеки libusb;*