

**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»**

студентки 2 курса, 24202 группы

Корсун Дарьи Андреевны

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:
Кандидат технических наук
В.А.Перепёлкин

Новосибирск 2025

СОДЕРЖАНИЕ

| | |
|----------------------|---|
| ЦЕЛЬ | 3 |
| ЗАДАНИЕ | 3 |
| ОПИСАНИЕ РАБОТЫ..... | 4 |
| ЗАКЛЮЧЕНИЕ | 6 |

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. За основу для разработки можно взять программу, приведенную в листинге 1.
2. Изучить состав и характеристики обнаруженных с помощью реализованной программ USB устройств.
3. Дополнить программу, реализованную в п. 2 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.

ОПИСАНИЕ РАБОТЫ

Для получения информации о доступных USB-устройствах с помощью библиотеки libusb. Ниже представлен код программы:

```
#include <libusb.h>

#include <iomanip>
#include <iostream>
#include <vector>

using namespace std;

void print_device_info(libusb_device *dev) {
    libusb_device_descriptor desc;
    int r = libusb_get_device_descriptor(dev, &desc);

    libusb_device_handle *handle = nullptr;
    libusb_open(dev, &handle);

    unsigned char buffer[256];

    if (desc.iSerialNumber == 0) {
        strcpy((char *)buffer, "unknown");
    } else {
        libusb_get_string_descriptor_ascii(handle, desc.iSerialNumber, buffer,
        sizeof(buffer)); }

    libusb_close(handle);

    cout << "Class: 0x" << hex << desc.bDeviceClass << endl;
    cout << "Vendor: 0x" << desc.idVendor << endl;
    cout << "Product: 0x" << desc.idProduct << dec << endl;
    cout << "Serial: " << string((char *)buffer) << endl;
}

int main() {
    libusb_context *ctx = nullptr;
    libusb_device **devs;
    ssize_t cnt;

    libusb_init(&ctx);
    libusb_set_debug(ctx, 3);
    cnt = libusb_get_device_list(ctx, &devs);

    cout << "Found USB devices: " << cnt << "\n" << endl;

    for (int i = 0; i < cnt; i++) {
        cout << i + 1 << ":" << endl;
        print_device_info(devs[i]);
    }

    libusb_free_device_list(devs, 1);
    libusb_exit(ctx);

    return 0;
}
```

Рассмотрим полученные устройства. Для изучения специально были подключены разные устройства для оценки выходных данных.

```
Class: 0x00          // код отсутствует
Vendor: 0x05ac      // Apple Inc
Product: 0x12a8      // Телефон
Serial: 00008130001225060A9A001C
```

```
Class: 0x00          // код отсутствует
Vendor: 0x05ac      // Apple Inc
Product: 0x1905      // Ноутбук
Serial: DF6Y63KQ1K
```

```
Class: 0x00          // код отсутствует
Vendor: 0x1d57      // Xenta
Product: 0x12a8
Serial: 00008130001225060A9A001C
```

```
Class: 0x09          // Концентратор
Vendor: 0x1d6b      // Linux Foundation
Product: 0x0003      // 3.0 root hub
Serial: 0000:04:00.4
```

```
Class: 0xef          // специф. устройство
Vendor: 0x0408      // Quanta Computer, Inc.
Product: 0x4033      // Model
Serial: 01.00.00
```

ЗАКЛЮЧЕНИЕ

В результате выполнения лабораторной работы было проведено ознакомление с началами низкоуровневого программирования периферийных устройств на примере получения информации о доступных USB-устройствах с помощью библиотеки libusb.