###### 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, 22202 группы

**Бальчинова А.С.**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

В.А. Перепёлкин

Новосибирск 2023

**СОДЕРЖАНИЕ**

[ЦЕЛЬ 3](#_Toc18443921)

[ЗАДАНИЕ 3](#_Toc18443922)

[ОПИСАНИЕ РАБОТЫ](#_Toc18443923) 3-16

[ЗАКЛЮЧЕНИЕ 16](#_Toc18443924)

# ЦЕЛИ

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.

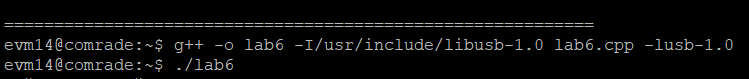
3. Дополнить программу функцией печати серийного номера USB устройства. Для написания функции рекомендуется использовать функции libusb\_open, libusb\_close, libusb\_get\_string\_descriptor\_ascii для печати поля iSerialNumber дескриптора устройства.

# ОПИСАНИЕ РАБОТЫ

1. Реализация программы на языке программирования C++.

#include <iostream>  
#include "libusb-1.0/libusb.h"  
#include <stdio.h>  
#include <clocale>  
#include <vector>  
#include <map>  
using namespace std;  
  
map<int, string> usbClasses = **{** {1, "аудио"},  
 {2, "коммуникационное устройство"},  
 {3, "устройство пользовательского интерфейса"},  
 {5, "физическое устройство"},  
 {6, "изображение"},  
 {7, "принтер"},  
 {8, "устройство хранения данных"},  
 {9, "hub"},  
 {10, "CDC-Data"},  
 {11, "Smart card"},  
 {13, "Content Security"},  
 {14, "видеоустройство"},  
 {15, "персональное медицинское устройство"},  
 {16, "аудио- и видеоустройства"},  
 {17, "Класс устройства Billboard"},  
 {18, "USB Type-C Bridge Class"},  
 {220, "диагностическое устройство"},  
 {224, "беспроводный контроллер"},  
 {239, "различные устройства"},  
 {254, "специфическое устройство"}  
**}**;  
  
void printdev(libusb\_device \*dev){  
 libusb\_device\_descriptor desc; // дескриптор устройства  
 libusb\_config\_descriptor \*config; // дескриптор конфигурации объекта  
 const libusb\_interface \*inter;  
 const libusb\_interface\_descriptor \*interdesc;  
 const libusb\_endpoint\_descriptor \*epdesc;  
 int r = libusb\_get\_device\_descriptor(dev, &desc);  
 if (r < 0){  
 fprintf(stderr, "Ошибка: дескриптор устройства не получен, код: %d.\n", r);  
 return;  
 }  
 libusb\_device\_handle\* handle = nullptr;  
 unsigned char str[256] = "";  
 int ret = libusb\_open(dev, &handle);  
 if (*LIBUSB\_SUCCESS* == ret) {  
 ret = libusb\_get\_string\_descriptor\_ascii(handle, desc.iProduct, str, 256);  
 if (ret > 0) {  
 cout << "Название: " << str <<endl;  
 }  
 ret = libusb\_get\_string\_descriptor\_ascii(handle, desc.iSerialNumber, str, 256);  
 if (ret > 0) {  
 cout << "Серийный номер: " << str <<endl;  
 }  
 }  
 // получить конфигурацию устройства  
 libusb\_get\_config\_descriptor(dev, 0, &config);  
 cout << "Число конфигураций: " << (int)desc.bNumConfigurations <<endl;  
 cout << "Класс устройства: " << usbClasses[(int)desc.bDeviceClass] <<endl;  
 cout << "id производителя: " << (int)desc.idVendor <<endl;  
 cout << "id устройства: " << (int)desc.idProduct <<endl;  
 cout << "Число интерфейсов: " << (int)config->bNumInterfaces <<endl;  
 for (int i = 0; i < (int)config->bNumInterfaces; i++) {  
 inter = &config->interface[i];  
 cout << "Интерфейс № " << i+1 << ":\n";  
 cout << "\t" << "Число альтернативных настроек: " << inter->num\_altsetting <<endl;  
 for(int j = 0; j < inter->num\_altsetting; j++) {  
 interdesc = &inter->altsetting[j];  
 cout << "\t\t" << "Альтернативные настройки № " << j+1 <<endl;  
 cout << "\t\t" << "Номер интерфейса: " << (int)interdesc->bInterfaceNumber <<endl;  
 cout << "\t\t" << "Класс интерфейса: " << usbClasses[(int)interdesc->bInterfaceClass] <<endl;  
 cout << "\t\t" << "Число конечных точек: " << (int)interdesc->bNumEndpoints <<endl;  
 for(int k = 0; k < (int)interdesc->bNumEndpoints; k++) {  
 epdesc = &interdesc->endpoint[k];  
 cout << "\t\t\t" << "Конечная точка № " << k+1 << ":" <<endl;  
 cout << "\t\t\t" << "Тип дескриптора: " << (int)epdesc->bDescriptorType <<endl;  
 cout << "\t\t\t" << "Адрес конечной точки: " << (int)epdesc->bEndpointAddress <<endl;  
 }  
 }  
 }  
 libusb\_free\_config\_descriptor(config);  
 libusb\_close(handle);  
}  
  
int main(){  
 setlocale(LC\_ALL, "Russian");  
 libusb\_device \*\*devs; //указатель на указатель на устройство, используется для получения списка устройств  
 libusb\_context \*ctx = nullptr; //контекст сессии libusb  
 int r; //для возвращаемых значений  
 ssize\_t cnt; //число найденных USB-устройств  
 ssize\_t i; //индексная переменная цикла перебора всех устройств  
 //инициализировать библиотеку libusb, открыть сессию работы с libusb  
 r = libusb\_init(&ctx);  
 if (r < 0){  
 fprintf(stderr, "Ошибка: инициализация не выполнена, код: %d.\n", r);  
 return 1;  
 }  
 // задать уровень подробности отладочных сообщений  
 libusb\_set\_option(ctx, *LIBUSB\_OPTION\_LOG\_LEVEL*, 3);  
 // получить список всех найденных USB- устройств  
 cnt = libusb\_get\_device\_list(ctx, &devs);  
 if (cnt < 0){  
 fprintf(stderr, "Ошибка: список USB устройств не получен, код: %d.\n", r);  
 return 1;  
 }  
 cout << "Найдено устройств: " << cnt <<endl;  
 for (i = 0; i < cnt; i++) { // цикл перебора всех устройств  
 cout << "Устройство: " << i+1 <<endl;  
 printdev(devs[i]); // печать параметров устройства  
 cout << "\n\n";  
 }  
 printf("===========================================================\n");  
 //освободить память, выделенную функцией получения списка устройств  
 libusb\_free\_device\_list(devs, 1);  
 libusb\_exit(ctx); //завершить работу с библиотекой libusb, закрыть сессию работы с libusb  
 return 0;  
}

1. Запуск программы и результат.

****

**Устройство: 1**

Серийный номер: 95410D90

Число конфигураций: 1

Класс устройства: различные устройства

id производителя: 1133

id устройства: 2085

Число интерфейсов: 4

**Устройство: 2**

Название: EHCI Host Controller

Серийный номер: 0000:00:1d.7

Число конфигураций: 1

Класс устройства: hub

id производителя: 7531

id устройства: 2

Число интерфейсов: 1

**Устройство: 3**

Название: UHCI Host Controller

Серийный номер: 0000:00:1d.2

Число конфигураций: 1

Класс устройства: hub

id производителя: 7531

id устройства: 1

Число интерфейсов: 1

**Устройство: 4**

Название: libusb: error: Permission denied

Серийный номер: error

Число конфигураций: 1

Класс устройства:

id производителя: 7247

id устройства: 38

Число интерфейсов: 2

**Устройство: 5**

Название: USB Optical Mouse

Число конфигураций: 1

Класс устройства:

id производителя: 1112

id устройства: 58

Число интерфейсов: 1

**Устройство: 6**

Название: UHCI Host Controller

Серийный номер: 0000:00:1d.1

Число конфигураций: 1

Класс устройства: hub

id производителя: 7531

id устройства: 1

Число интерфейсов: 1

**Устройство: 7**

Название: UHCI Host Controller

Серийный номер: 0000:00:1d.0

Число конфигураций: 1

Класс устройства: hub

id производителя: 7531

id устройства: 1

Число интерфейсов: 1

**Устройство: 8**

Название: USB2.0-CRW

Серийный номер: 20060413092100000

Число конфигураций: 1

Класс устройства:

id производителя: 3034

id устройства: 385

Число интерфейсов: 1

**Устройство: 9**

Название: EHCI Host Controller

Серийный номер: 0000:00:1a.7

Число конфигураций: 1

Класс устройства: hub

id производителя: 7531

id устройства: 2

Число интерфейсов: 1

**Устройство: 10**

Название: UHCI Host Controller

Серийный номер: 0000:00:1a.2

Число конфигураций: 1

Класс устройства: hub

id производителя: 7531

id устройства: 1

Число интерфейсов: 1

**Устройство: 11**

Название: UHCI Host Controller

Серийный номер: 0000:00:1a.1

Число конфигураций: 1

Класс устройства: hub

id производителя: 7531

id устройства: 1

Число интерфейсов: 1

**Устройство: 12**

Название: UHCI Host Controller

Серийный номер: 0000:00:1a.0

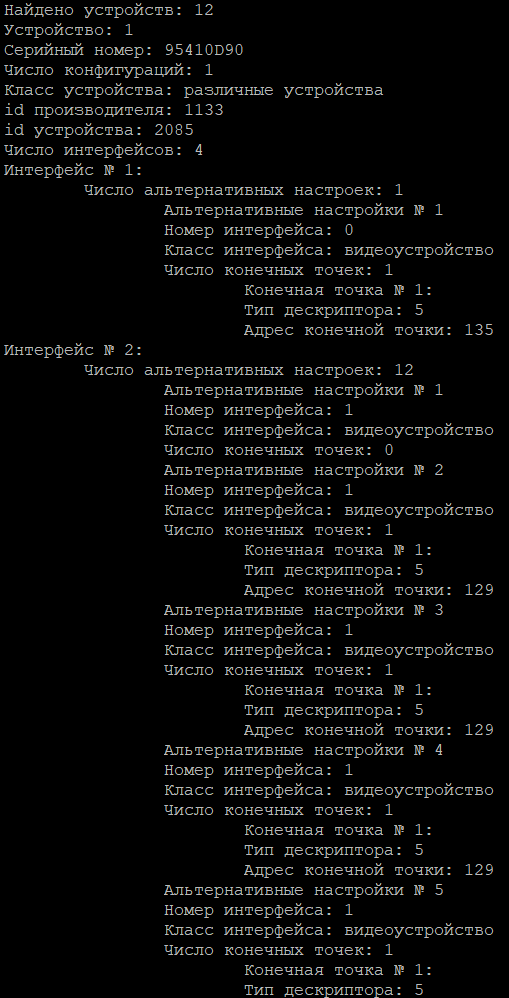
Число конфигураций: 1

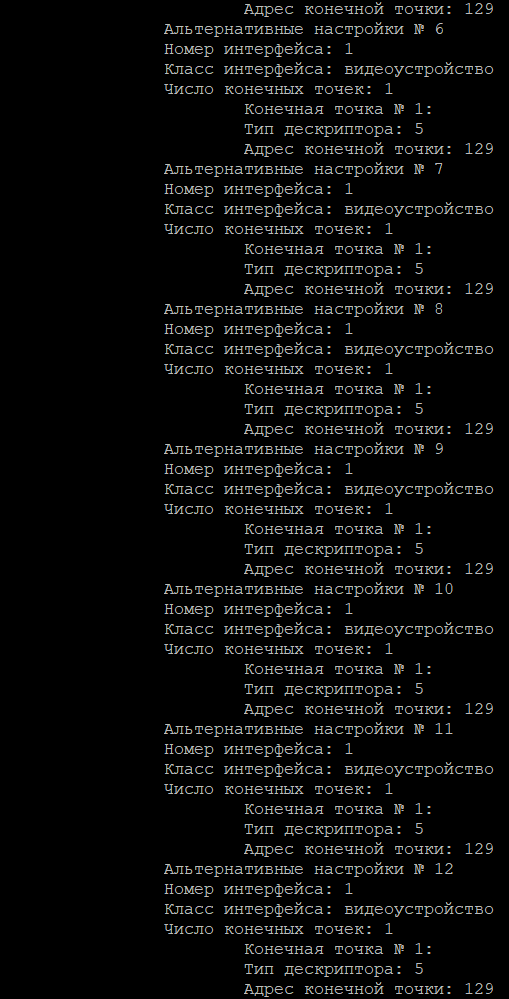
Класс устройства: hub

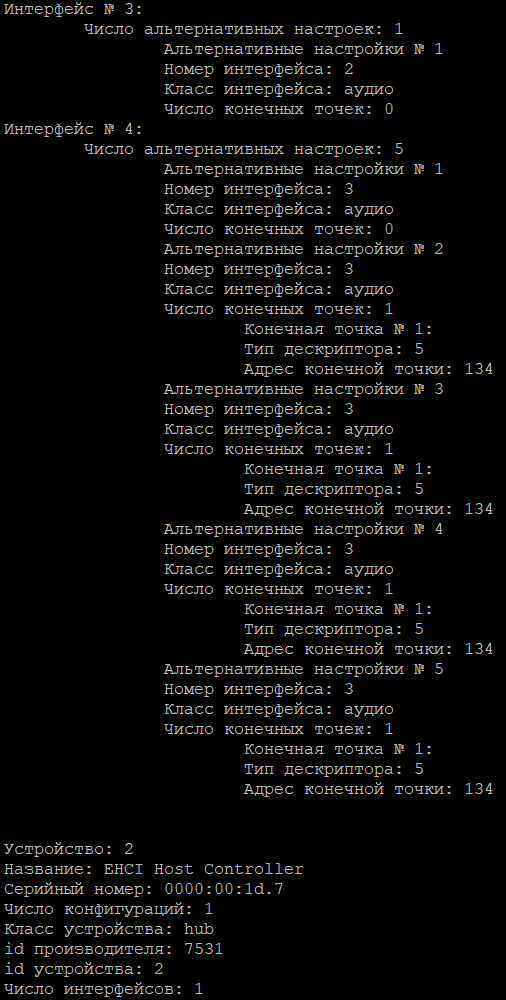
id производителя: 7531

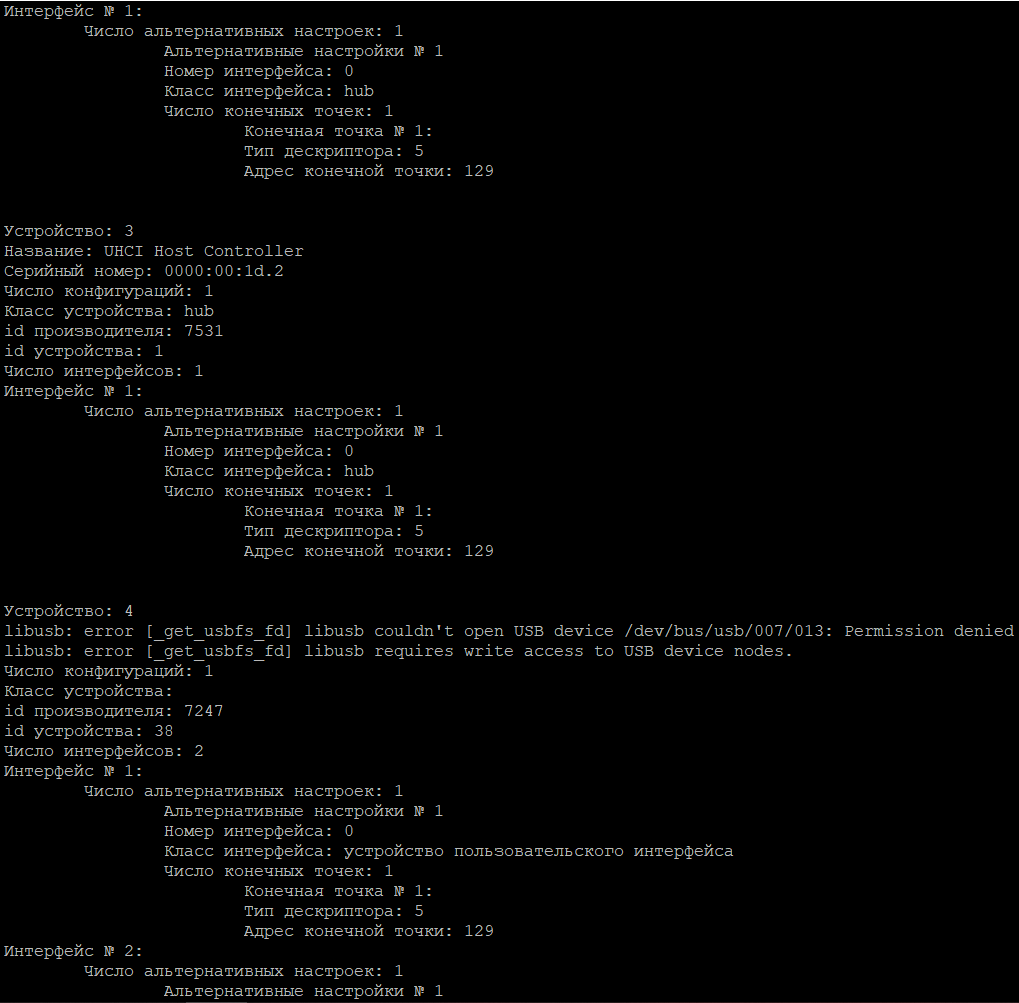
id устройства: 1

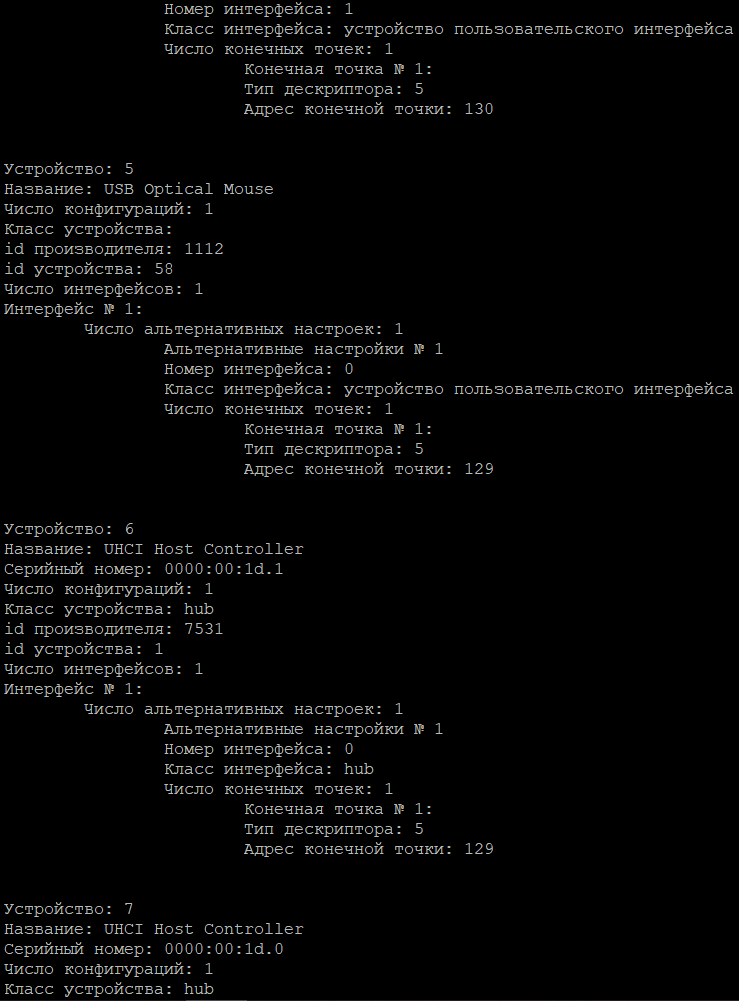
Число интерфейсов: 1

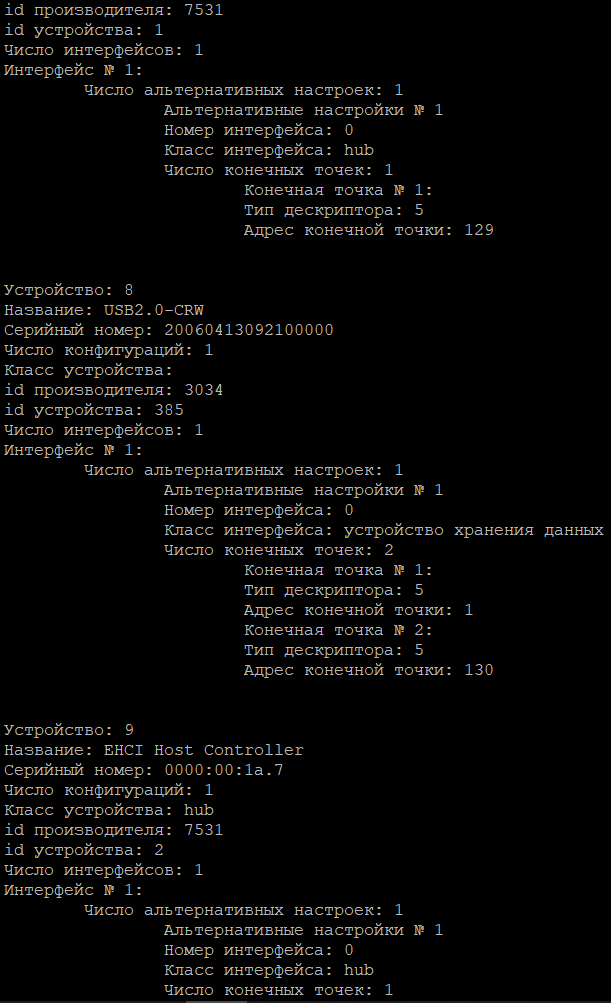
****

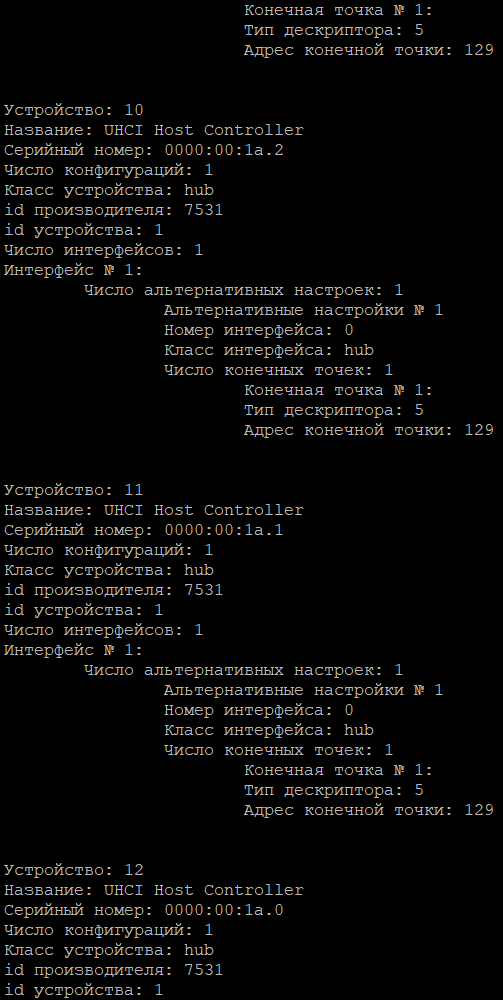
****

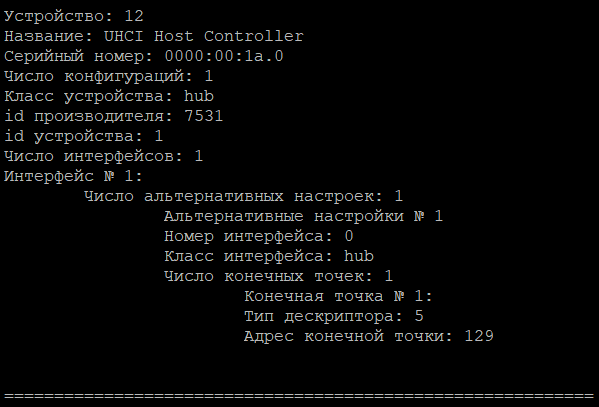












# ЗАКЛЮЧЕНИЕ

Ознакомились с началами низкоуровневого программирования периферийных устройств на примере получения информации о доступных USB-устройствах с помощью библиотеки libusb.