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, 18209 группы

**Большим Максима Антоновича**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

Власенко Андрей Юрьевич

Новосибирск 2019

**Оглавление:**

[**Цель**](#_rcx03e8skstc) **3**

[**Задание**](#_gjdgxs) **3**

[**Листинг программы**](#_s3v8rfijkd4k) **3**

[**Описание обнаруженных USB-устройств**](#_6pxgbunyfiif) **4**

[**Заключение**](#_qajn46z3wr1g) **6**

# Цель

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

# Задание

Изучить принципы низкоуровневой работы с USB-устройствами с помощью библиотеки libusb. Ознакомиться с тем, как происходит работа с девайсами и какие данные хранит сам девайс. Проанализировать работу кода и сделать выводы на основе результата работы.

# Листинг программы

#include <libusb-1.0/libusb.h>

#include <cstdio>

void printData(libusb\_device \*dev);

int main()

{

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 countUSBDevices; // число найденных USB-устройств

r = libusb\_init(&ctx); // инициализировать библиотеку libusb, открыть сессию работы с libusb

if (r < 0)

{

fprintf(stderr,"Ошибка: инициализация не выполнена, код: %d.\n", r);

return 1;

}

// получить список всех найденных USB- устройств

countUSBDevices = libusb\_get\_device\_list(ctx, &devs);

if (countUSBDevices < 0)

{

fprintf(stderr,

"Ошибка: список USB устройств не получен. Код: %d\n", r);

return 1;

}

printf("найдено устройств: %ld\n", countUSBDevices);

printf("===========================================================\n");

printf("|\* класс устройства\n");

printf("| | \* идентификатор производителя\n");

printf("| | | \* идентификатор устройства\n");

printf("| | | | \* серийный номер\n");

printf("+--+----+----+--------------------+\n");

for (ssize\_t i = 0; i < countUSBDevices; i++)

printData(devs[i]); // печать параметров устройства

printf("===========================================================\n");

libusb\_free\_device\_list(devs, 1); // освободить память, выделенную функцией получения списка устройств

libusb\_exit(ctx); // завершить работу с библиотекой libusb, закрыть сессию работы с libusb

return 0;

}

void printData(libusb\_device \*dev)

{

libusb\_device\_descriptor desc{}; // дескриптор устройства

libusb\_device\_handle \*handle = nullptr; // хэндл устройства

unsigned char str[256]; // строка для хранения серийного номера

int r = libusb\_get\_device\_descriptor(dev, &desc); // получить дескриптор

if (r < 0)

{

fprintf(stderr,"Ошибка: дескриптор устройства не получен, код: %d.\n", r);

return;

}

printf(" %.2x %.4x %.4x ",

(int) desc.bDeviceClass,

desc.idVendor,

desc.idProduct

);

libusb\_open(dev, &handle);

if (handle && desc.iSerialNumber)

{

r = libusb\_get\_string\_descriptor\_ascii(handle, desc.iSerialNumber, str, sizeof(str));

if (r > 0)

printf("%s", str);

else

printf("empty");

} else

printf("null");

putchar('\n');

}

# Описание обнаруженных USB-устройств

Программа, запущенная на ноутбуке, где проводилась работа, вывела следующие устройства:

найдено устройств: 10

===========================================================

|\* класс устройства

| | \* идентификатор производителя

| | | \* идентификатор устройства

| | | | \* серийный номер

+--+----+----+--------------------+

09 8087 8000 null

09 1d6b 0002 null

09 8087 8008 null

09 1d6b 0002 null

09 1d6b 0003 null

ff 0bda 0139 null

ef 04f2 b3fd null

e0 8087 07dc null

00 046d c52f null

09 1d6b 0002 null

===========================================================

Выведенные устройства:

1) 09 8087 8000 null

Данное устройство является USB-контроллером. Intel Corp.

2) 09 1d6b 0002 null

Данное устройство является незанятым usb 2.0 и поэтому обозначается как Linux Foundation 2.0 root hub.

3) 09 8087 8008 null

Данное устройство является USB-контроллером. Intel Corp.

4) 09 1d6b 0002 null

Данное устройство является незанятым usb 2.0 и поэтому обозначается как Linux Foundation 2.0 root hub.

5) 09 1d6b 0003 null

Данное устройство является незанятым usb 3.0 и поэтому обозначается как Linux Foundation 3.0 root hub.

6) ff 0bda 0139 null

Realtek Semiconductor Corp. RTS5139 Card Reader Controller. Это устройство — кард-ридер старых носителей данных.

7) ef 04f2 b3fd null

Chicony Electronics Co., Ltd HD WebCam (Asus N-series). Веб-камера ноутбука.

8) e0 8087 07dc null

Данное устройство является USB-контроллером. Intel Corp.

9) 00 046d c52f null

Logitech, Inc. Unifying Receiver. Блютуз-ресивер для беспроводной мыши.

10) 09 1d6b 0002 null

Данное устройство является незанятым usb 2.0 и поэтому обозначается как Linux Foundation 2.0 root hub.

Как можно заметить, ни у какой модели нету серийного номера. Как показывает практика — наличие серийного номера никак не гарантировано, т.к. девайсы:

1) Системные устройства не в нём не нуждаются;

2) Пиратское изделие;

3) Повреждённые данные на устройстве или намеренно стёртые.

# Заключение

В ходе работы с libusb были освоены:

1) работа с usb-устроствами;

2) получение данных с usb-устройств и оперирование над ними.

Так же появилось понимание тех процессов, что происходят на низких уровнях. Благодаря новым знаниям можно предполагать, почему новое устройство не подключается и затем решать проблему. Стоит отметить, что такое можно проводить и системными устройствами, т.к. некоторые из них подключены к usb-шине.