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, группы 19201  
  
Хаецкая Дарья Владимировна

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

Власенко Андрей Юрьевич

Новосибирск 2020

# 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 <iostream>

#include <libusb-1.0/libusb.h>

#include <stdio.h>

#include <sys/types.h>

using namespace std;

void printdev(libusb\_device \*dev);

int main(){

libusb\_device \*\*devs; // указатель на указатель на устройство,

// используется для получения списка устройств

libusb\_context \*ctx = NULL; // контекст сессии libusb

int r; // для возвращаемых значений

ssize\_t cnt; // число найденных USB-устройств

ssize\_t i; // индексная переменная цикла перебора всех 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;

}

// задать уровень подробности отладочных сообщений

libusb\_set\_debug(ctx, 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.\n", r);

return 1;

}

printf("найдено устройств: %d\n", cnt);

printf("=============================="

"=============================\n");

printf("\* количество возможных конфигураций\n");

printf("| \* класс устройства\n");

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

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

printf("+--+--+----+------------------"

"---+--+----+-----------------\n");

for(i = 0; i < cnt; i++) { // цикл перебора всех устройств

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

}

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;

}

void printInfo(libusb\_device \*dev, int\* r, libusb\_device\_descriptor\* desc){

unsigned char string[256];

libusb\_device\_handle \*handler; // хендлер где будем хранить конфигурации

libusb\_open(dev, &handler);

cout << endl;

if (desc->iSerialNumber) {

// получить дескриптор устройства в виде строки символов

\*r = libusb\_get\_string\_descriptor\_ascii(handler, desc->iSerialNumber, string, sizeof(string));

if (\*r > 0)

printf("SerialNumber: %s\n", string);

}

if (desc->iManufacturer) {

\*r = libusb\_get\_string\_descriptor\_ascii(handler, desc->iManufacturer, string, sizeof(string));

if (\*r > 0)

printf("Manufacturer: %s\n", string);

}

if (desc->iProduct) {

\*r = libusb\_get\_string\_descriptor\_ascii(handler, desc->iProduct, string, sizeof(string));

if (\*r > 0)

printf("Product: %s\n", string);

}

cout << endl;

libusb\_close(handler);

}

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); // Get the USB device descriptor for a given device.

if (r < 0){

fprintf(stderr,

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

return;

}

libusb\_get\_config\_descriptor(dev, 0, &config); // получить конфигурацию устройства

printf("=============================="

"=============================\n");

printf("%.2d %.2d %.4d %.4d\n", (int)desc.bNumConfigurations,

(int)desc.bDeviceClass, desc.idVendor, desc.idProduct, (int)config->bNumInterfaces);

printInfo(dev, &r, &desc);

libusb\_free\_config\_descriptor(config);

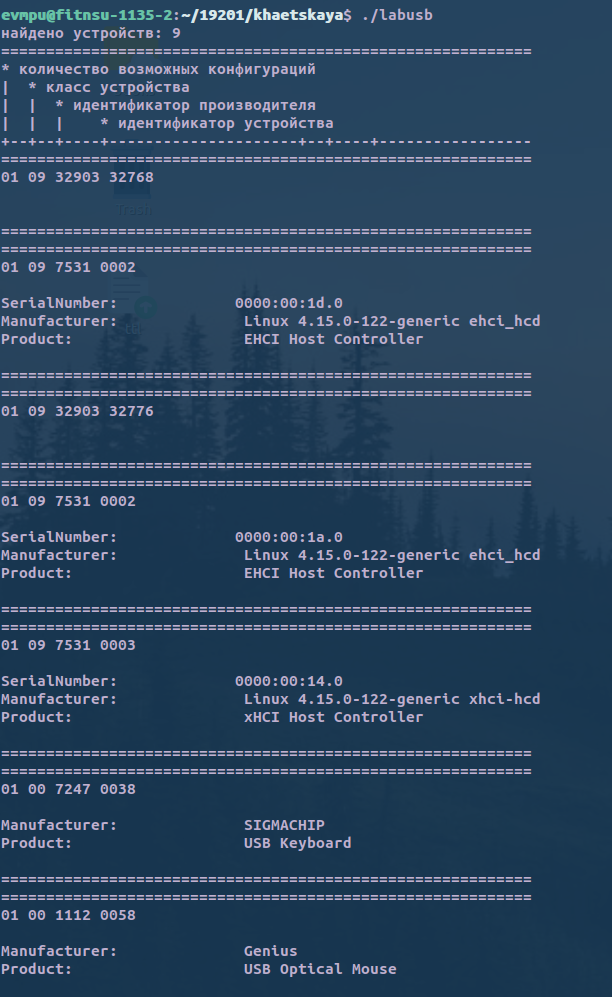
printf("=============================="

"=============================\n");

}

# 

# Описание обнаруженных USB-устройств

Программа, запущенная на сервере, вывела следующие устройства:  


# 

# 

# 

# Заключение

В ходе работы с libusb были освоены:

1) работа с usb-устройствами;

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