

**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, 23209 группы
Инокова Семёна Шухратовича

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:
А. Ю. Кудинов

Новосибирск 2024

СОДЕРЖАНИЕ

ЦЕЛЬ.....	3
ЗАДАНИЕ.....	3
ОПИСАНИЕ РАБОТЫ.....	4
ЗАКЛЮЧЕНИЕ.....	5
Приложение 1. <i>Код программы для работы</i>	6
Приложение 2. <i>Вывод программы</i>	8

ЦЕЛЬ

1. 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, 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.

ОПИСАНИЕ РАБОТЫ

1. Был написан код на языке C++, который выводит следующие данные:
 - Количество найденных устройств.
 - Класс устройства.
 - ID производителя.
 - ID изделия.
 - Серийный номер.
 - Производитель.
 - Модель устройства.
2. Код скомпилирован с помощью команды:
 - `g++ -o executable3 -I/usr/include/libusb-1.0 main3.cpp -lusb-1.0`

ЗАКЛЮЧЕНИЕ

В ходе данной работы я познакомился с библиотекой libusb, с помощью которой научился выводить информацию о доступных USB-устройствах.

Приложение 1. Код программы для работы

```
#include <iostream>
#include <libusb.h>
#include <stdio.h>

using namespace std;

void print_device_info(libusb_device *device, libusb_context *ctx);

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

    r = libusb_init(&ctx);
    if (r < 0) {
        fprintf(stderr, "Ошибка: инициализация не выполнена, код: %d.\n", r);
        return 1;
    }
    libusb_set_option(ctx, LIBUSB_OPTION_LOG_LEVEL, LIBUSB_LOG_LEVEL_NONE);

    cnt = libusb_get_device_list(ctx, &devs);
    if (cnt < 0) {
        fprintf(stderr, "Ошибка: список USB устройств не получен.\n");
        libusb_exit(ctx);
        return 1;
    }

    printf("Найдено устройств: %zd\n", cnt);
    printf("| Класс устройства | ID производителя | ID изделия | Серийный номер |\n");
    printf("-----\n");
    for (ssize_t i = 0; i < cnt; ++i) {
        libusb_device *device = devs[i];
        libusb_device_descriptor desc;

        r = libusb_get_device_descriptor(device, &desc);
        if (r < 0) {
            fprintf(stderr, "Ошибка: дескриптор устройства не получен, код: %d.\n", r);
            continue;
        }

        printf("| %16d | %15x | %10x |",
               desc.bDeviceClass,
               desc.idVendor,
               desc.idProduct);

        print_device_info(device, ctx);
        printf("-----\n");
    }

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

    return 0;
}
```

```

void print_device_info(libusb_device *device, libusb_context *ctx) {
    libusb_device_handle *handle;
    libusb_device_descriptor desc;
    unsigned char serial_number[256];
    unsigned char manufacturer[256];
    unsigned char product[256];
    int r;

    r = libusb_get_device_descriptor(device, &desc);
    if (r < 0 || desc.iSerialNumber == 0) {
        printf(" Нет серийного номера |\n");
        return;
    }

    r = libusb_open(device, &handle);
    if (r < 0) {
        printf(" Ошибка открытия устройства |\n");
        return;
    }

    r = libusb_get_string_descriptor_ascii(handle, desc.iSerialNumber, serial_number, sizeof(serial_number));
    if (r > 0) {
        printf(" %s |\n", serial_number);
    } else {
        printf(" Не удалось получить серийный номер |\n");
    }

    if (desc.iManufacturer) {
        r = libusb_get_string_descriptor_ascii(handle, desc.iManufacturer, manufacturer, sizeof(manufacturer));
        if (r > 0) {
            printf("Производитель: %s\n", manufacturer);
        } else {
            printf("Производитель: Не удалось получить информацию.\n");
        }
    } else {
        printf("Производитель: Не указан.\n");
    }

    if (desc.iProduct) {
        r = libusb_get_string_descriptor_ascii(handle, desc.iProduct, product, sizeof(product));
        if (r > 0) {
            printf("Модель устройства: %s\n", product);
        } else {
            printf("Модель устройства: Не удалось получить информацию.\n");
        }
    } else {
        printf("Модель устройства: Не указана.\n");
    }

    libusb_close(handle);
}

```

Приложение 2. Вывод программы

```
evm116@comrade:~/lab6$ ./main
Найдено устройств: 12
| Класс устройства | ID производителя | ID изделия | Серийный номер |
-----|-----|-----|-----|
|          239 |          46d |          825 | 95410D90 |
Производитель: Не указан.
Модель устройства: Не указана.
-----|-----|-----|-----|
|          9 |          1d6b |          2 | 0000:00:1d.7 |
Производитель: Linux 5.15.0-125-generic ehci_hcd
Модель устройства: EHCI Host Controller
-----|-----|-----|-----|
|          9 |          1d6b |          1 | 0000:00:1d.2 |
Производитель: Linux 5.15.0-125-generic uhci_hcd
Модель устройства: UHCI Host Controller
-----|-----|-----|-----|
|          0 |          1c4f |          26 | Нет серийного номера |
-----|-----|-----|-----|
|          0 |          458 |          3a | Нет серийного номера |
-----|-----|-----|-----|
|          9 |          1d6b |          1 | 0000:00:1d.1 |
Производитель: Linux 5.15.0-125-generic uhci_hcd
Модель устройства: UHCI Host Controller
-----|-----|-----|-----|
|          9 |          1d6b |          1 | 0000:00:1d.0 |
Производитель: Linux 5.15.0-125-generic uhci_hcd
Модель устройства: UHCI Host Controller
-----|-----|-----|-----|
|          0 |          bda |          181 | 20060413092100000 |
Производитель: Generic
Модель устройства: USB2.0-CRW
-----|-----|-----|-----|
|          9 |          1d6b |          2 | 0000:00:1a.7 |
Производитель: Linux 5.15.0-125-generic ehci_hcd
Модель устройства: EHCI Host Controller
-----|-----|-----|-----|
|          9 |          1d6b |          1 | 0000:00:1a.2 |
Производитель: Linux 5.15.0-125-generic uhci_hcd
Модель устройства: UHCI Host Controller
-----|-----|-----|-----|
|          9 |          1d6b |          1 | 0000:00:1a.1 |
Производитель: Linux 5.15.0-125-generic uhci_hcd
Модель устройства: UHCI Host Controller
-----|-----|-----|-----|
|          9 |          1d6b |          1 | 0000:00:1a.0 |
Производитель: Linux 5.15.0-125-generic uhci_hcd
Модель устройства: UHCI Host Controller
-----|-----|-----|-----|
```