АППАРАТНОЕ РАЗРЕШЕНИЕ ЗАПУСКА ПРИЛОЖЕНИЯ ПРИ ПОДКЛЮЧЕНИИ УСТРОЙСТВА В ПОРТ

Утилита для ОС Windows

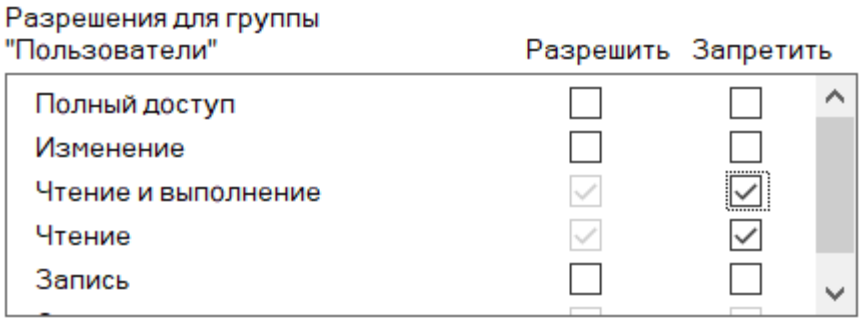
Жемелев Г. А., гр. 13541/3

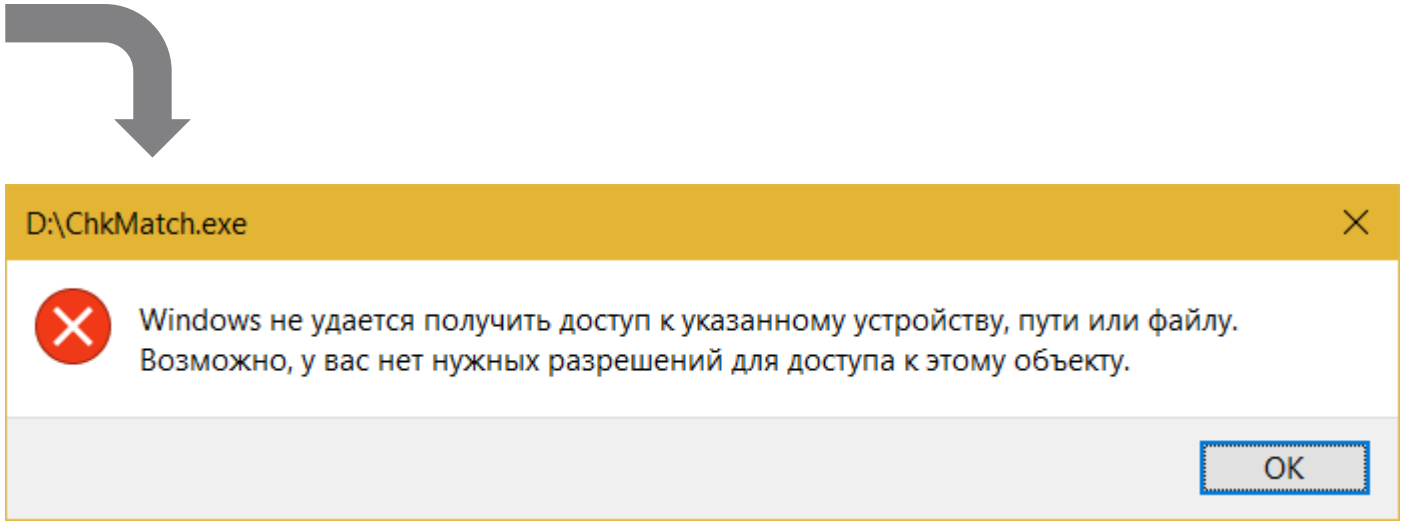
Что на входе?

- Наличие зарегистрированного утилитой USB-устройства
- Это устройство подключено к USB-порту
- Полное имя файла для блокировки доступа к нему
- Наличие прав для управления доступом к файлам пользователя

Что на выходе?

- Запрет на чтение и выполнение указанного файла
- При вставке устройства – запрет должен быть снят





Что необходимо для реализации?

Работа с USB

- Идентификация подключенных устройств
- Вывод списка подключенных устройств пользователю
- Хранение идентификатора выбранного пользователем устройства
- Реакция на подключение или отключение устройств

Контроль доступа

- Установка запрета на чтение и выполнение (в т. ч. на чтение)
- Снятие запрета на чтение и выполнение (в т. ч. на чтение)
- Пользователь не должен иметь возможности самостоятельно вернуть себе доступ во время действия блокировки

ПРОЕКТИРОВАНИЕ

Программного комплекса USB Locker

Состав программного комплекса

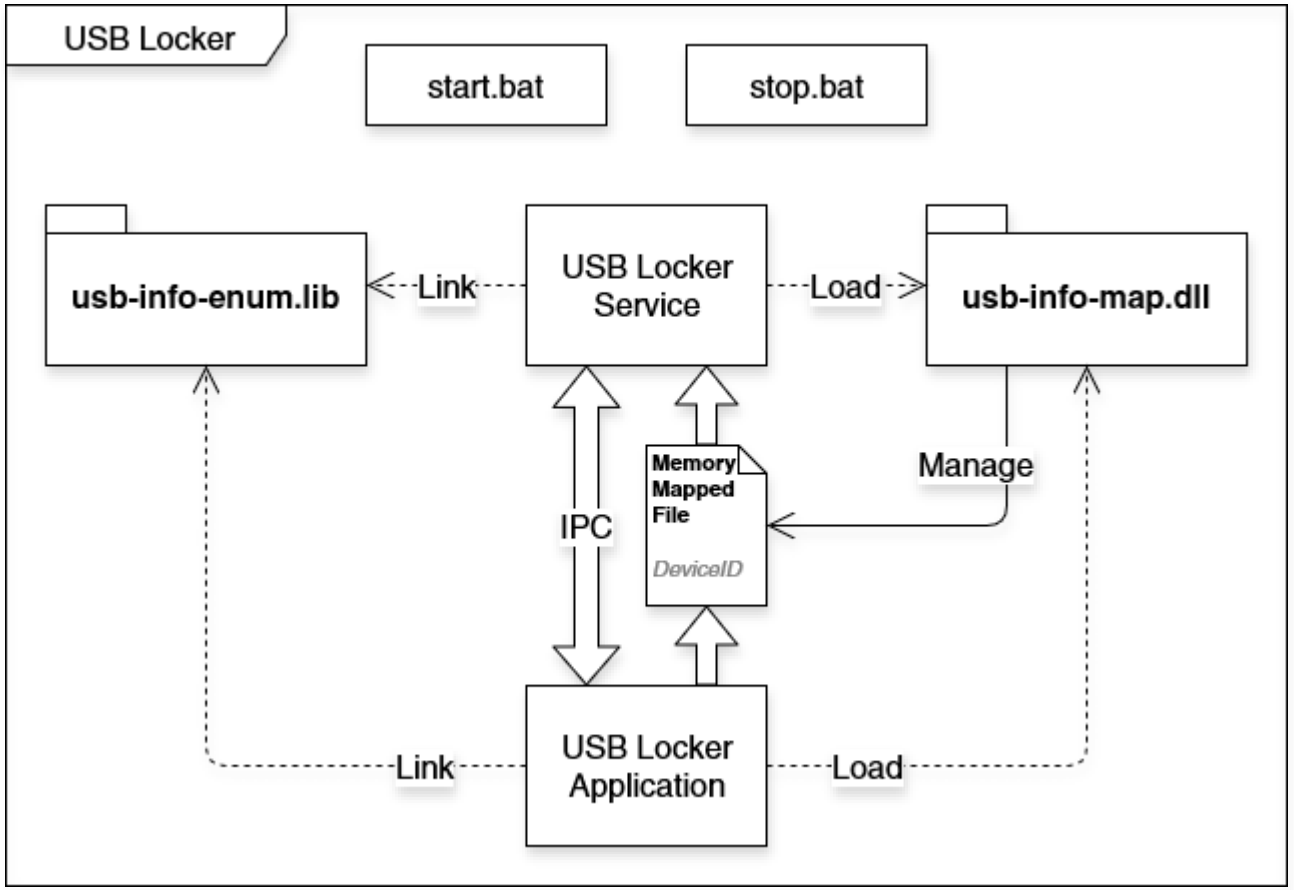
Две программы: консольное приложение и служба – а также скрипт запуска:

- **Приложение USB Locker** предоставляет пользователю интерфейс для выбора одного устройству из списка подключенных в данный момент USB-устройств.
- **Служба USB Locker Service** работает в фоновом режиме и отслеживает наличие заданного USB-устройства в списке подключенных, управляя доступом к файлу.
- **Скрипт запуска** принимает на вход полное имя файла для контроля доступа к нему, устанавливает службу (если она не установлена) и передаёт ей имя файла, после чего запускает приложение USB Locker.

Структура программного комплекса

Служба и приложение в составе утилиты USB Locker используют:

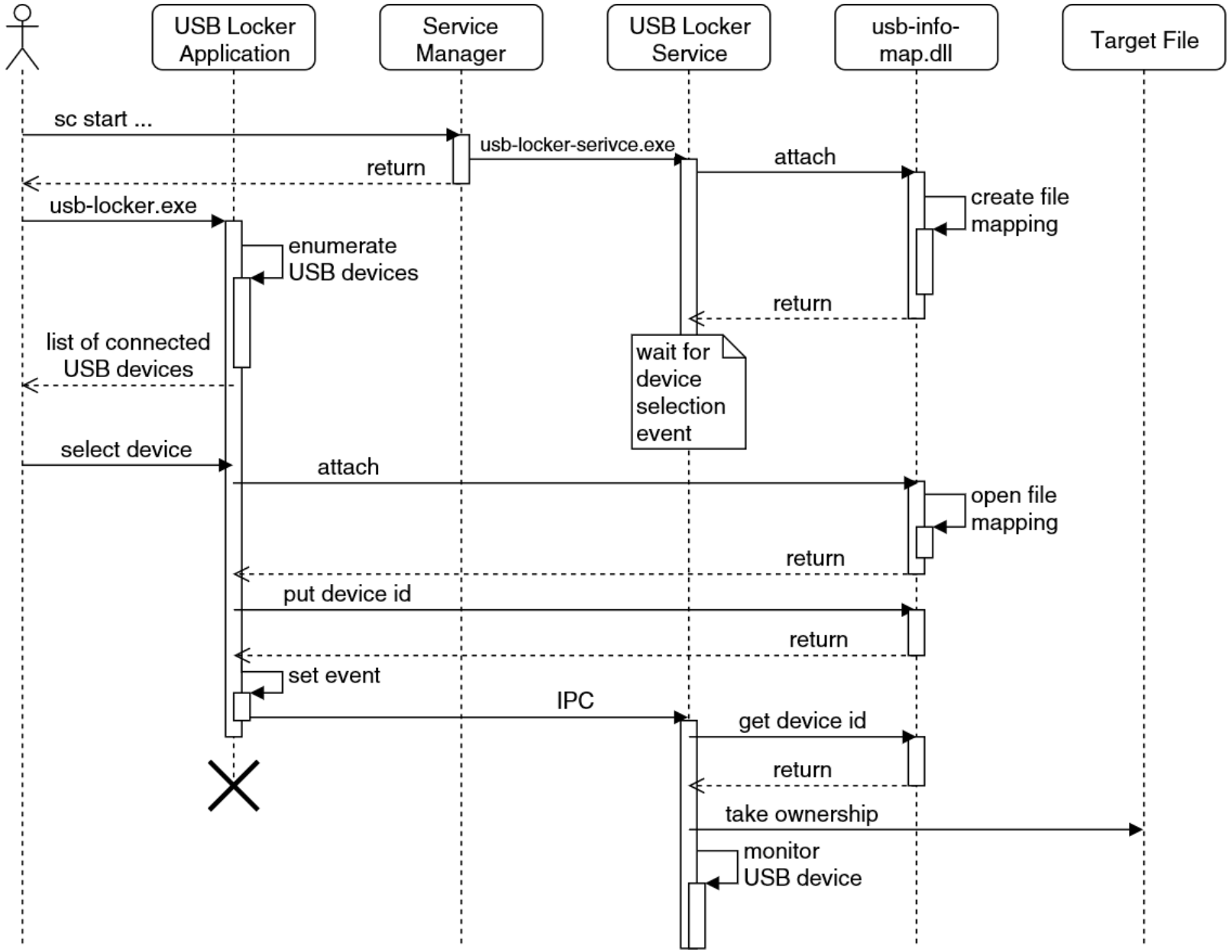
- **статическую библиотеку** usb-info-enum.lib для работы с USB-устройствами;
- **отображенный в память файл** для передачи идентификатора выбранного устройства;
- **динамическую библиотеку** usb-info-map.dll для управления этим файлом.



Взаимодействие компонентов: *выбор устройства*

Служба USB Locker Service уже установлена.

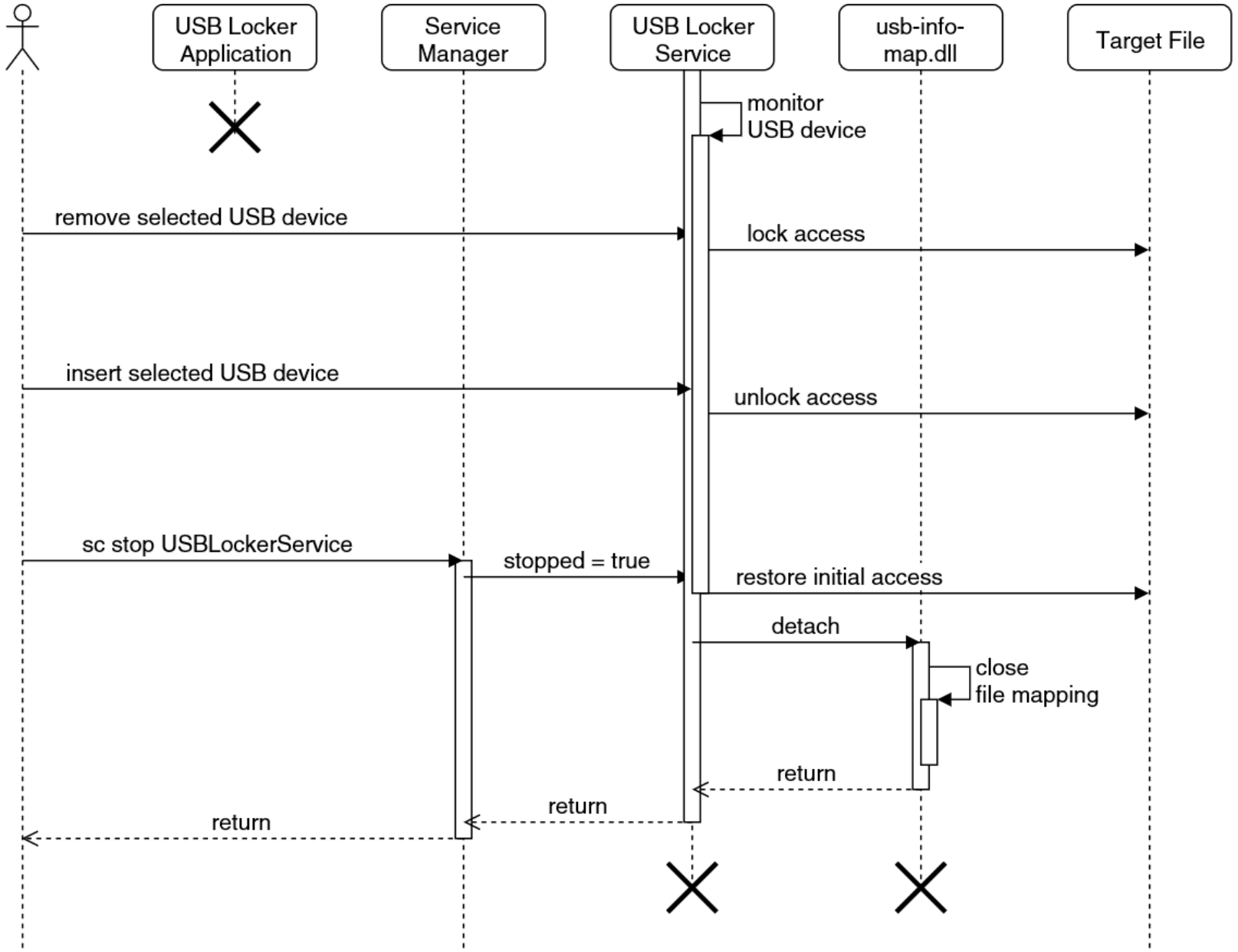
Отсоединение USB Locker Application от динамической библиотеки usb-info-map.dll не показано для компактности.



Взаимодействие компонентов: *контроль доступа*

При остановке службы целевому файлу возвращаются первоначальный дескриптор безопасности.

Библиотека usb-info-map.dll закрывает отображение файла, когда от неё отсоединяется последний процесс.



РЕАЛИЗАЦИЯ

Программного комплекса USB Locker

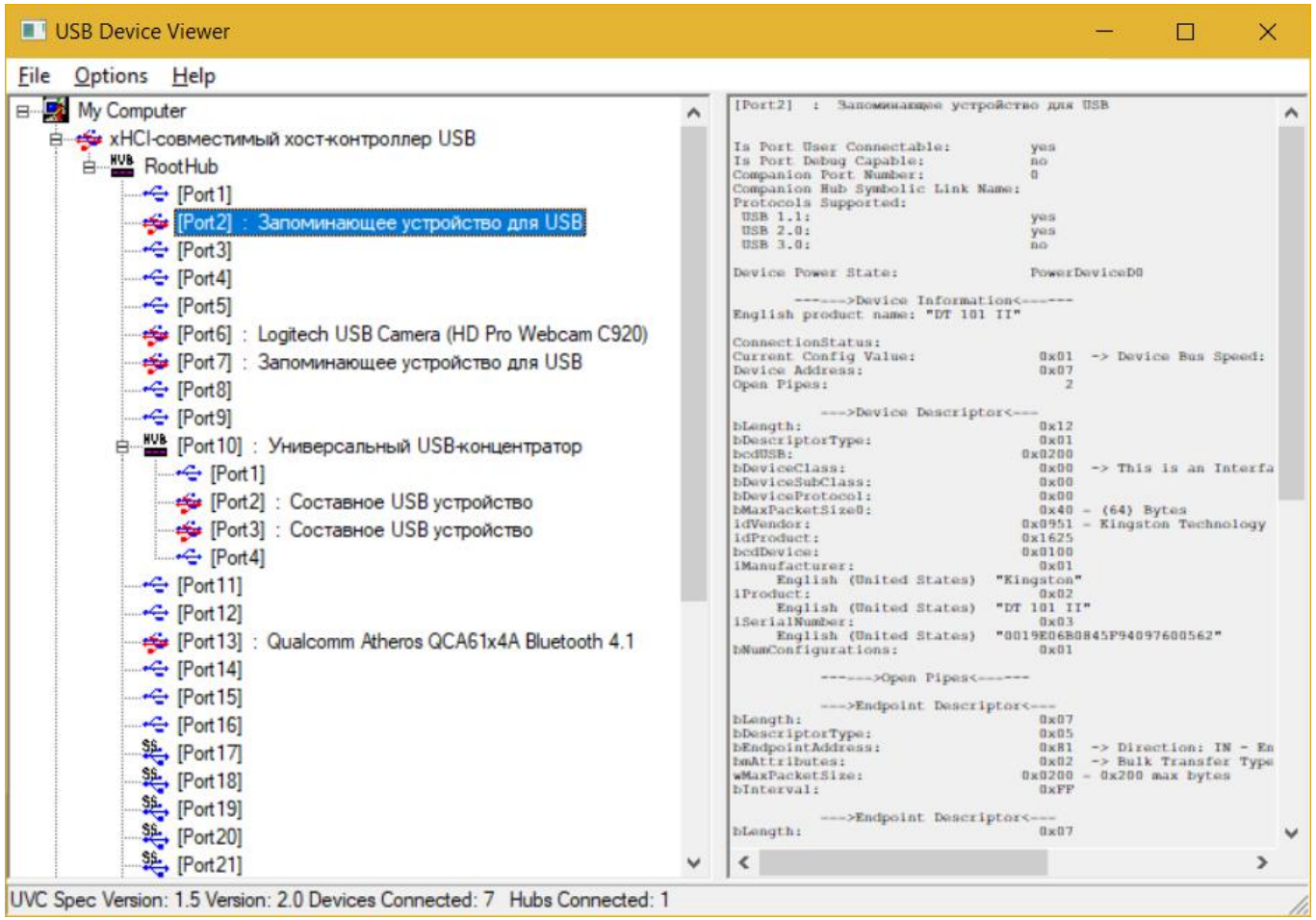
РАБОТА С USB-УСТРОЙСТВАМИ

Создание статической библиотеки usb-info-enum.lib

USB View

- Утилита в составе Windows SDK
- Имеет открытый исходный код
- Выводит информацию обо всех USB-устройствах

Итоги анализа:
установлены функции WinAPI, позволяющие выполнить перечисление USB-устройств.



https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/usbview
https://github.com/Microsoft/Windows-driver-samples/tree/master/usb/usbview

Перечисление (enumeration) устройств

Функции системной библиотеки SetupAPI:

- **SetupDiGetClassDevs** – получение описателя (handle) структуры DEVINFO, хранящей описание всех устройств заданного класса.
- **SetupDiEnumDeviceInfo** – обход списка устройств из DEVINFO с записью информации о текущем устройстве в структуру SP_DEVINFO_DATA.
- **SetupDiGetDeviceProperty** – получение из структуры SP_DEVINFO_DATA сведений об устройстве по ключу типа DEVPROPKEY. Интересующие ключи:
  - *DEVPKEY_Device_FriendlyName* (понятное имя устройства)
  - *DEVPKEY_Device_DeviceDesc* (понятное описание устройства)
  - *DEVPKEY_Device_ContainerId* (идентификатор контнйнера)

Идентификация реальных USB-устройств

- Начиная с ОС Windows 7, для устройств был введен новый тип идентификаторов, называемый **Container ID – идентификатор контейнера**, – с тем чтобы позволить системе обнаруживать многофункциональные устройства, т.е. физические устройства, которым соответствует множество логических устройств.
- Идентификатор контейнера **представляется в виде GUID** и возвращается как одноименная структура:

```
typedef struct _GUID {
    unsigned long  Data1;
    unsigned short Data2;
    unsigned short Data3;
    unsigned char  Data4[ 8 ];
} GUID;
```

Идентификация реальных USB-устройств

Алгоритм получения максимально полной информации о USB-устройствах:

1. Получить список A всех подключенных устройств.
2. Получить список B всех подключенных USB-устройств.
3. Извлечь из списка A все элементы, Container ID которых содержится среди элементов списка B.
4. Объединить все элементы с одинаковым Container ID путем конкатенации их понятных имен и понятных описаний.

На выходе – набор структур вида «устройство – набор описаний».

Идентификация реальных USB-устройств

Используемые структуры:

```
struct DeviceInfo {
    wstring friendlyName;
    wstring description;
    const wstring toWideString() const;
    const string toString() const;
};
```

```
struct DeviceDescription {
    Guid guid;
    DeviceInfo info;
};
```

```
struct RealDevice {
    Guid guid;
    vector<DeviceInfo> incarnations;
};
```

Идентификация реальных USB-устройств

Функция для получения списка реальных USB-устройств:

```
const vector<RealDevice> composeRealDevices(vector<DeviceDescription> allDevices,
        unordered_set<Guid> usbDevicesIds) {
    vector<RealDevice> realDevices;
    for each (const Guid& realDeviceId in usbDevicesIds) {
        RealDevice realDevice = { realDeviceId, vector<DeviceInfo>() };
        for each (const DeviceDescription& logicalDevice in allDevices) {
            if (logicalDevice.guid == realDeviceId)
                realDevice.incarnations.push_back(logicalDevice.info);
        }
        realDevices.push_back(realDevice);
    }
    return realDevices;
}
```

Идентификация реальных USB-устройств

C:\WINDOWS\system32\cmd.exe

```
Список подключенных USB-устройств:

1) Устройство с кодом контейнера {3369C49B-807C-11E7-898A-806E6F6E696}, известное системе как:
     > HID-совместимый системный контроллер
     > Logitech Unifying USB receiver
     > Составное USB устройство
     > HID-совместимое устройство, определенное поставщиком
     > HID-совместимое устройство, определенное поставщиком
     > HID-совместимая мышь
     > Клавиатура HID
     > HID-совместимое устройство, определенное поставщиком
     > USB Input Device (Logitech Download Assistant)
     > HID-совместимое устройство управления
     > Logitech HID-compliant Unifying device
     > USB-устройство ввода

2) Устройство с кодом контейнера {B519715E-2B25-5EB2-A7B6-D9F63EBA380}, известное системе как:
     > Запоминающее устройство для USB
     > Kingston DT 101 II USB Device (Дисковый накопитель)
     > KINGSTON (DT 101 II       )
     > Том

Для продолжения нажмите любую клавишу . . .
```

Библиотека **usb-info-enum.lib**

Итоговый API из функций C++ для получения информации об устройствах:

```
const bool checkIfConnected(GUID containerId);
const vector<RealDevice> getExistingUsbDevices();
const vector<RealDevice> composeRealDevices(vector<DeviceDescription> allDevices,
       unordered_set<Guid> ids);
const vector<DeviceDescription> enumerateAllDevicesInfo();
const vector<DeviceDescription> enumerateUSBDevicesInfo();
const vector<DeviceDescription> enumerateDevicesInfo(const GUID* classGuid, DWORD
       scope);
const wstring getDevicePropertyAsWideString(HDEVINFO hDevInfoSet, PSP_DEVINFO_DATA
       devInfoData, const DEVPROPKEY* key);
const GUID getDevicePropertyAsGUID(HDEVINFO hDevInfoSet, PSP_DEVINFO_DATA
       devInfoData, const DEVPROPKEY* key);
```

ИСПОЛЬЗОВАНИЕ ОТОБРАЖАЕМОГО В ПАМЯТЬ ФАЙЛА В КАЧЕСТВЕ СРЕДСТВА IPC

Создание динамической библиотеки usb-info-map.dll

Отображение файла в память

Использованные функции WinAPI для работы с отображенным в память файлом:

- **CreateFileMapping** – создание именованного объекта отображения или получение его описателя, если объект с таким именем уже существует.
- **MapViewOfFile** – отображение файла в адресное пространство вызывающего процесса по описателю объекта отображения. Возвращаемое значение – указатель на начало отображенного файла.
- **CopyMemory** – простой способ записи структуры GUID в отображенный файл.
- **UnmapViewOfFile** – освобождение памяти, выделенной под отображенный файл в адресном пространстве вызывающего процесса.

Отображение файла в память

Вызов CreateFileMapping:

```
hMappedFile = CreateFileMapping(
   INVALID_HANDLE_VALUE,    // use paging file
   &securityAttr,           // specific security
   PAGE_READWRITE,          // read/write access
   0,                       // maximum object size (high-order DWORD)
   BUF_SIZE,                // maximum object size (low-order DWORD)
   SHM_NAME);               // name of mapping object
```

При создании отображаемого файла используются атрибуты безопасности, полученные из дескриптора безопасности, который был создан путем вызова:

```
ConvertStringSecurityDescriptorToSecurityDescriptor(
   TEXT("D:(A;;FA;;;BA)"), SDDL_REVISION_1, &pMappedFileSD, NULL);
```

Отображение файла в память

Вызов MapViewOfFile:

```
pFileBuffer = (LPBYTE) MapViewOfFile(hMappedFile, FILE_MAP_ALL_ACCESS, 0, 0, BUF_SIZE);
```

Помещение и извлечение идентификатора контейнера целевого устройства:

```
API const GUID getUsbDeviceIdentifier() {
   GUID deviceId;
   CopyMemory((LPVOID)&deviceId, pFileBuffer, sizeof(GUID));
   return deviceId;
}

API void putUsbDeviceIdentifier(GUID deviceId) {
   CopyMemory(pFileBuffer, (LPVOID)&deviceId, sizeof(GUID));
}
```

Библиотека **usb-info-map.dll**

Для использования в USB Locker необходимо экспортировать только две функции, которые были представлены на предыдущем слайде.

Директива `API` определена следующим образом:

```
#ifdef USBINFOMAP_EXPORTS
#define API extern "C" __declspec(dllexport)
#else
#define API extern "C" __declspec(dllimport)
#endif
```

Имя и размер отображаемого в память файла:

```
#define SHM_NAME L"Global\\USBLockerSHM"
#define BUF_SIZE sizeof(GUID)
```

Библиотека **usb-info-map.dll**

Использование возможностей точки входа DllMain для управления созданием/удалением отображаемого в память файла:

```
BOOL APIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID lpReserved) {
   switch (ul_reason_for_call) {
   case DLL_PROCESS_ATTACH:
      return tryToAttach(); // ← увеличение счетчика подключенных процессов и контроль ошибок,
   case DLL_THREAD_ATTACH:  //   информация об ошибках записывается в лог-файл
      return TRUE;
   case DLL_THREAD_DETACH:
      return TRUE;
   case DLL_PROCESS_DETACH:
      detach();    // ← уменьшение счетчика подключенных процессов и освобождение памяти при
      return TRUE; //   достижении его нулевого значения
   }
   return FALSE;
}
```

Библиотека **usb-info-map.dll**

Всё это дает возможность максимально простого использования созданной библиотеки. Пример из USB Locker Application:

```
void placeDeviceIdIntoMappedFile(const Guid& selectedDeviceId) {
   HINSTANCE mapperLib = LoadLibrary(L"usb-info-map.dll");
   if (mapperLib == NULL) {
      wcout << L"Не удалось загрузить usb-info-map.dll (код ошибки 0x" <<
          hex << GetLastError() << L")" << endl;
      exit(1);
   }
   auto putUsbDeviceIdentifier = (void(*)(GUID))
      GetProcAddress(mapperLib, "putUsbDeviceIdentifier");
   putUsbDeviceIdentifier(selectedDeviceId.guid);
   if (mapperLib != NULL) FreeLibrary(mapperLib);
}
```

ВЗАИМОДЕЙСТВИЕ С ПОЛЬЗОВАТЕЛЕМ

Создание приложения USB Locker

Алгоритм работы USB Locker Application

1. Получить список реальных USB-устройств и пронумеровать их.
2. Вывести полученный список на экран.
3. Запросить у пользователя выбор одного из устройств или осуществить выход при вводе символа «q» (без учета регистра).
4. Если ввод осуществлен корректно, записать идентификатор выбранного устройства в отображенный в память файл.
5. «Разбудить» ожидающую службу USB Locker Service с помощью механизма событий, предоставляемого ОС Windows.

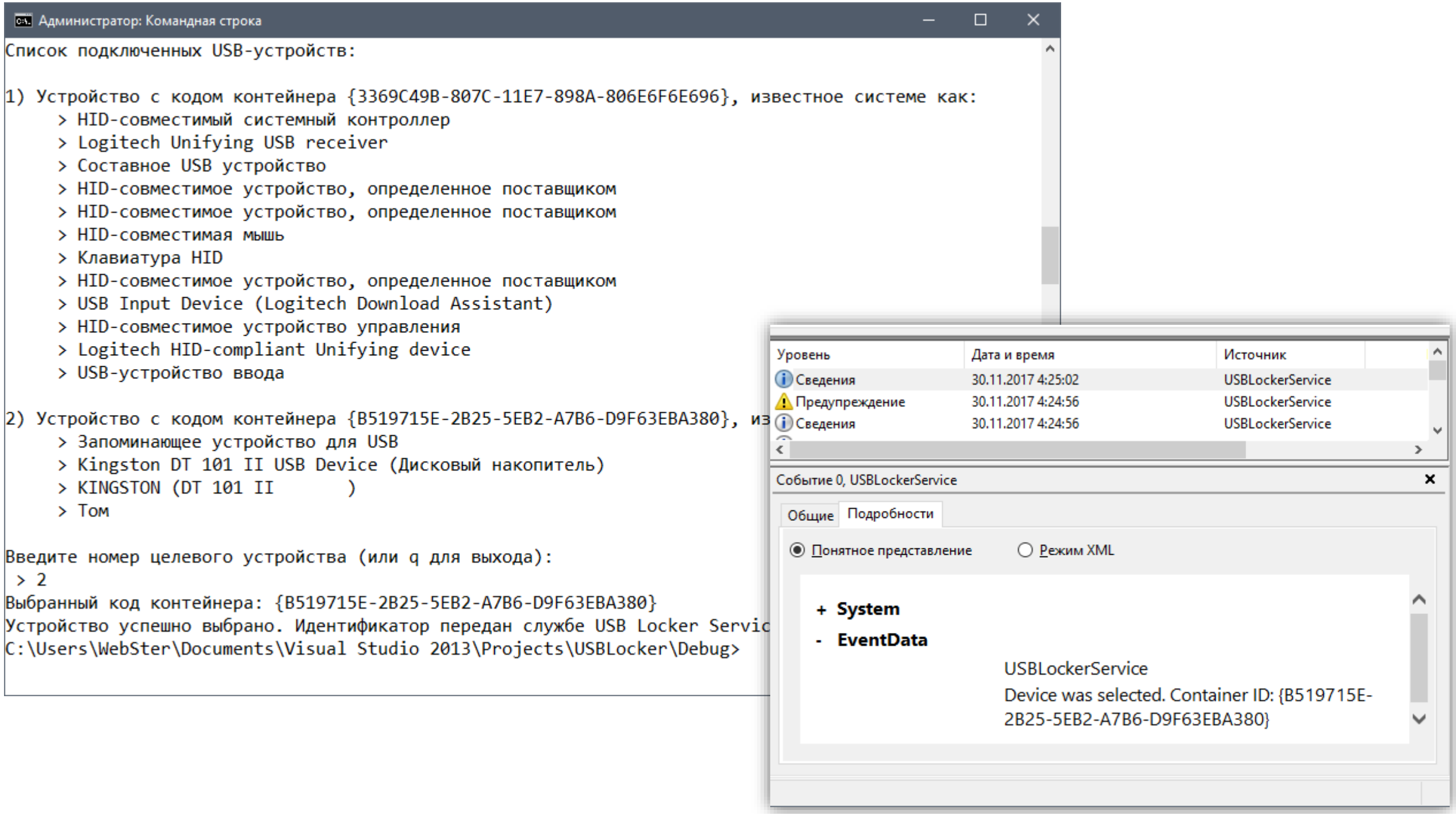
Описанные шаги составляют функцию *main*.

Использование IPC в USB Locker Application

Данное приложение использует два средства IPC:

- разделяемая память (с помощью библиотеки usb-info-map.dll),
- события (events).

```
void wakeUpLockerService() {
   HANDLE hEvent = OpenEvent(EVENT_MODIFY_STATE, FALSE, L"Global\\USBLockerDeviceSelectEvent");
   if (hEvent == NULL) {
      if (GetLastError() == ERROR_FILE_NOT_FOUND) {
         wcout << L"Невозможно продолжить работу программы: "
            L"проверьте, что служба USB Locker Service запущена." << endl;
      }
      exit(1);
   }
   SetEvent(hEvent);
   CloseHandle(hEvent);
}
```


Администратор: Командная строка
Список подключенных USB-устройств:
1) Устройство с кодом контейнера {3369C49B-807C-11E7-898A-806E6F6E696}, известное системе как:
> HID-совместимый системный контроллер
> Logitech Unifying USB receiver
> Составное USB устройство
> HID-совместимое устройство, определенное поставщиком
> HID-совместимое устройство, определенное поставщиком
> HID-совместимая мышь
> Клавиатура HID
> HID-совместимое устройство, определенное поставщиком
> USB Input Device (Logitech Download Assistant)
> HID-совместимое устройство управления
> Logitech HID-compliant Unifying device
> USB-устройство ввода
2) Устройство с кодом контейнера {B519715E-2B25-5EB2-A7B6-D9F63EBA380}, из
> Запоминающее устройство для USB
> Kingston DT 101 II USB Device (Дисковый накопитель)
> KINGSTON (DT 101 II)
> Том
Введите номер целевого устройства (или q для выхода):
> 2
Выбранный код контейнера: {B519715E-2B25-5EB2-A7B6-D9F63EBA380}
Устройство успешно выбрано. Идентификатор передан службе USB Locker Servic
C:\Users\WebSter\Documents\Visual Studio 2013\Projects\USBLocker\Debug>
Уровень
Дата и время
Источник
Сведения
30.11.2017 4:25:02
USBLockerService
Предупреждение
30.11.2017 4:24:56
USBLockerService
Сведения
30.11.2017 4:24:56
USBLockerService
Событие 0, USBLockerService
Общие
Подробности
Понятное представление
Режим XML
+ System
- EventData
USBLockerService
Device was selected. Container ID: {B519715E-2B25-5EB2-A7B6-D9F63EBA380}

КОНТРОЛЬ ДОСТУПА К ФАЙЛУ И МОНИТОРИНГ USB-УСТРОЙСТВА

Создание службы USB Locker Service

Состав приложения-службы

Служба USB Locker Service содержит следующие классы:

- **ServiceBase** – стандартный базовый класс службы.
- **ServiceInstaller** – стандартный класс установщика службы.
- **USBLockerService** – класс, наследующий ServiceBase и реализующий требуемую функциональность программы USB Locker Service.
- **FileMapper** – класс, инкапсулирующий работу с отображенным в память файлом.
- **DeviceMonitor** – класс для мониторинга состояния выбранного USB-устройства.
- **AccessLocker** – класс для контроля доступа к файлу.

Стандартные классы:
https://code.msdn.microsoft.com/windowsapps/CppWindowsService-cacf4948/

Особенности программирования служб

- Служба – циклично выполняющееся приложение, не имеющее пользовательского интерфейса. Как следствие, об ошибках и других событиях необходимо сообщать посредством **системного журнала**.
- Служба должна своевременно **отвечать на запросы менеджера служб** (Service Manager), поэтому недопустимо «вечное» ожидание и требуется периодическая проверка необходимости завершить выполнение основного цикла программы.
- Служба выполняется в отдельной сессии, поэтому объекты IPC для взаимодействия с пользовательскими приложениями должны находится в **глобальном пространстве имен**.
- При запуске службы от имени LocalSystem **атрибуты безопасности** по умолчанию для создаваемых объектов необходимо ослаблять.

Атрибуты безопасности для создаваемых объектов IPC

Атрибуты, назначаемые по умолчанию, не дают доступ даже встроенной группе «Администраторы». Поэтому необходимо ослабить их:

```
BOOL success = ConvertStringSecurityDescriptorToSecurityDescriptor(
   TEXT("D:(A;;FA;;;BA)"), SDDL_REVISION_1, &pEventsSD, NULL);
if (!success) {
   throw std::system_error(GetLastError(), std::system_category(),
      "Could not create a security descriptor");
}
SECURITY_ATTRIBUTES securityAttr = asSecurityAttributes(pEventsSD);

// Создание объекта события, которое будет использовано после выбора устройства
hDeviceSelectionEvent = createEvent(DEVICE_SELECT_EVENT_NAME, &securityAttr);
```

```
void USBLockerService::ServiceWorkerThread(void) {
   try {
      FileMapper fileMapper;
      waitForDeviceSelection();
      if (!m_fStopping) {
         GUID deviceId = fileMapper.getUSBDeviceIdentifier();
         logInfo(L"Device was selected. Container ID: " + guidToWideString(deviceId));
         DeviceMonitor usbMonitor(deviceId);
         CThreadPool::QueueUserWorkItem(&DeviceMonitor::listenForEvents, &usbMonitor);
         AccessLocker locker(targetFile);
         while (!m_fStopping) {
            // <...> Основной цикл
         }
         usbMonitor.stop = true;
         locker.restoreTargetFileSD();
      }
   } catch (const std::system_error& err) {
      std::wostringstream ss;
      ss << SERVICE_NAME << L" failed with error: " << err.what()
         << L"(error_code " << err.code() << L")";
      WriteEventLogEntry(ss.str().c_str(), EVENTLOG_ERROR_TYPE);
   }
   // Signal the stopped event.
   SetEvent(m_hStoppedEvent);
}
```

Основной цикл службы USB Locker Service

```
DWORD result = WaitForSingleObject(hUsbEvent , STOP_CHECK_PERIOD_MS);
switch (result) {
case WAIT_OBJECT_0:
    // hUsbEvent signaled
    ResetEvent(hUsbEvent);
    if (usbMonitor.isConnected() && locker.isLocked()) {
        locker.unlock();
        logInfo(L"File unlocked!");
    } else if (!usbMonitor.isConnected() && !locker.isLocked()) {
        locker.lock();
        logInfo(L"File locked!");
    }
    break;
case WAIT_TIMEOUT:
    // End loop iteration
    break;
default:
    throw GetLastError();
}
```

Мониторинг состояния USB-устройства

Для того чтобы определить наличие или отсутствия подключенного устройства можно использовать функцию *checkIfConnected* из статической библиотеки **usb-info-enum.lib**:

```
void DeviceMonitor::listenForEvents() {
   bool previous_state = connected;
   bool current_state = connected;
   while (!stop) {
      current_state = checkIfConnected(targetDeviceContainerId);
      if (previous_state != current_state) {
         connected = current_state;
         SetEvent(hUsbEvent);
         previous_state = current_state;
      }
      Sleep(1000);
   }
}
```

Работа с отображенным в память файлом

Благодаря тому что библиотека **usb-info-map.dll** сама следит за используемыми ресурсами, получение идентификатора выбранного устройства реализуется очень просто.

```
FileMapper::FileMapper() {
    mapperLib = LoadLibrary(L"usb-info-map.dll");
    if (mapperLib == NULL) {
        throw GetLastError();
    }
}

FileMapper::~FileMapper() {
    if (mapperLib != NULL) {
        FreeLibrary(mapperLib);
    }
}

const GUID FileMapper::getUSBDeviceIdentifier() {
    auto getUsbDeviceIdentifier = (const GUID(*)())
        GetProcAddress(mapperLib, "getUsbDeviceIdentifier");
    return getUsbDeviceIdentifier();
}
```

Установка прав доступа к файлам

Программная реализация – путем использования **Access Control Lists** (ACL):

- идентифицировать объект для установки прав (в данном случае - файл)
- получить его текущий Discretionary Access Control List (DACL)
- *сделать резервную копию DACL* (восстановление – при остановке службы)
- добавить в DACL запись (Access Control Entry) с указанием пользователя и прав
- прикрепить обновленный DACL к объекту

Источник: "Modifying the ACLs of an Object in C++" (MSDN)
http://msdn.microsoft.com/en-us/library/aa379283(v=VS.85).aspx

Контроль доступа к целевому файлу

Управлять доступом к файлу могут пользователи, наделенный привилегией SE_TAKE_OWNERSHIP_NAME, а также *владелец* самого *файла*.

Чтобы предотвратить возможность владельца вернуть себе доступ к файлу, необходимо программно **сменить владельца целевого файла** на пользователя «Администратор».

Так как по умолчанию привилегия SE_TAKE_OWNERSHIP_NAME не активна, то для этого необходимо использовать следующие функции:

- **OpenProcessToken** – получение токена (access token) текущего процесса.
- **LookupPrivilegeValue** – получение локального идентификатора привилегии.
- **AdjustTokenPrivileges** – активация привилегии для указанного токена.
- **SetNamedSecurityInfo** – установка атрибутов безопасности для файла.

Контроль доступа к целевому файлу

Для того чтобы переключаться между запретом и разрешением на доступ пользователя к файлу, можно хранить в классе AccessLocker два разных DACL – один с разрешающей Access Control Entry, другой – с запрещающей:

```
pLockedACL = constructACL(MODIFIED_PERMISSIONS, DENY_ACCESS, { NULL,
NO_MULTIPLE_TRUSTEE, TRUSTEE_IS_SID, TRUSTEE_IS_WELL_KNOWN_GROUP,
(LPTSTR)pSIDEveryone });

pUnlockedACL = constructACL(MODIFIED_PERMISSIONS, SET_ACCESS, { NULL,
NO_MULTIPLE_TRUSTEE, TRUSTEE_IS_SID, TRUSTEE_IS_WELL_KNOWN_GROUP,
(LPTSTR)pSIDEveryone });
```

Контроль доступа к целевому файлу

```
PACL AccessLocker::constructACL(DWORD permissions, ACCESS_MODE mode, TRUSTEE trustee) {
    PACL pACL = NULL;
    EXPLICIT_ACCESS ea[1];
    ZeroMemory(&ea, sizeof(EXPLICIT_ACCESS));

    ea[0].grfAccessPermissions = permissions;
    ea[0].grfAccessMode = mode;
    ea[0].grfInheritance = NO_INHERITANCE;
    ea[0].Trustee = trustee;

    DWORD dwRes = SetEntriesInAcl(NUM_ACES, ea, NULL, &pACL);
    if (dwRes != ERROR_SUCCESS) {
        throw std::system_error(GetLastError(), std::system_category(),
            "Failed to SetEntriesInAcl");
    }
    return pACL;
}
```

SDDL

Уже после реализации конструирования DACL, приведенной выше, был найден значительно более короткий и простой способ – использование **языка определения дескрипторов безопасности SDDL** (Security Descriptor Definition Language).

Запись всего дескриптора в виде одной строки из условных обозначений:

O:owner_sid
G:group_sid
D:dacl_flags(string_ace1)(string_ace2)… (string_acen)
S:sacl_flags(string_ace1)(string_ace2)… (string_acen)

Источник: https://msdn.microsoft.com/en-us/library/aa379567(v=vs.85).aspx

SDDL

Язык SDDL был использован для создания дескрипторов безопасности для средств IPC:

```
ConvertStringSecurityDescriptorToSecurityDescriptor(
   TEXT("D:(A;;FA;;;BA)"), SDDL_REVISION_1, &pEventsSD, NULL);
```

Приведенная строка на языке SDDL означает следующее:

«Дескриптор безопасности, в списке DACL (D:) *которого есть одна запись ACL: выдать* (A) *пользователю «Администраторы»* (BA) *доступ на полный файловый контроль* (FA)*».*

Скрипты запуска и остановки

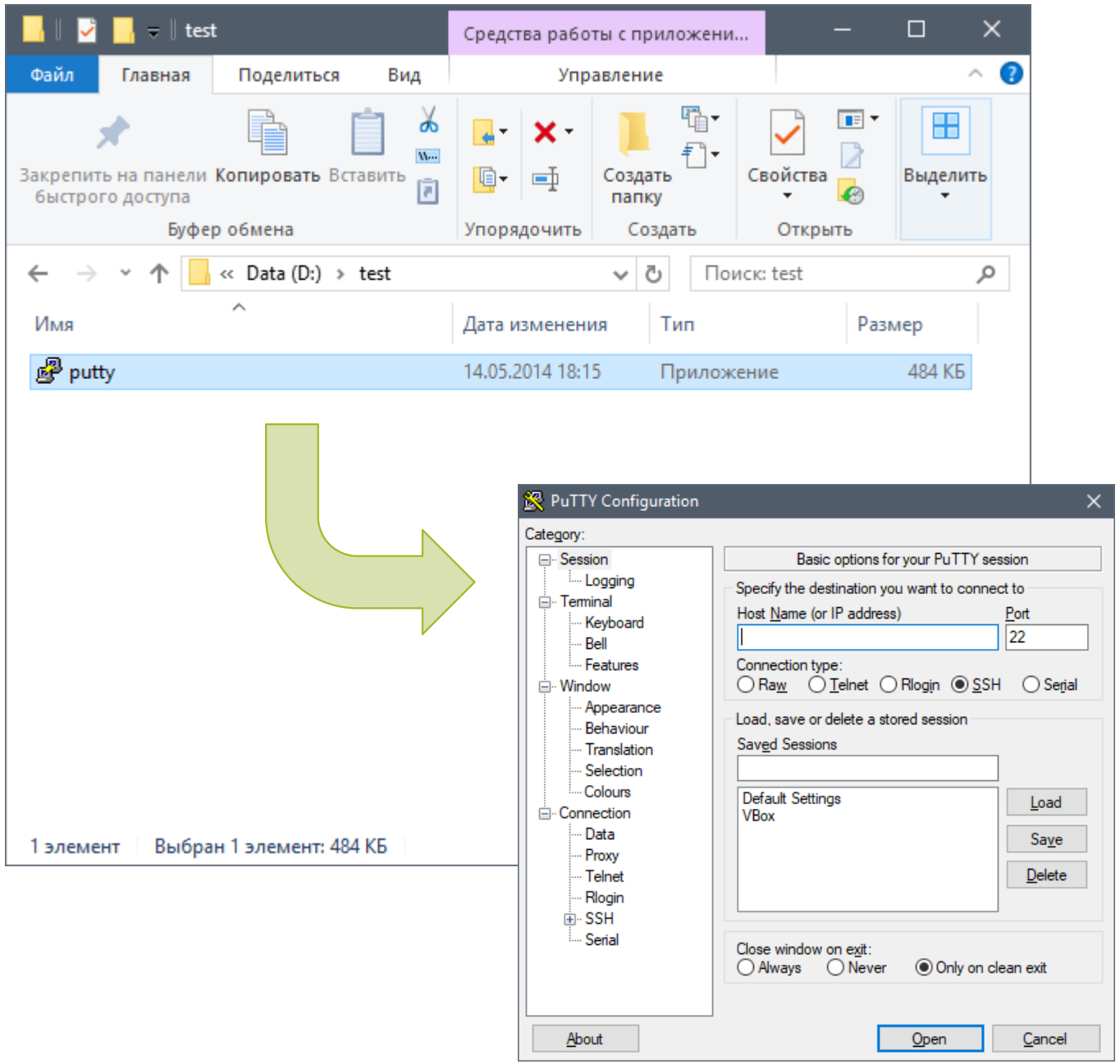
Для упрощения запуска и остановки службы были написаны **скрипты start.bat и stop.bat**. Они используют команды менеджера служб и вызывают приложения из разработанного программного комплекса.

Скрипт *start.bat* принимает в качестве единственного аргумента имя файла, который подлежит блокировке. Если служба USB Locker Service не установлена, то он устанавливает её, а затем запускает, передавая имя файла. После этого запускается приложение USB Locker для выбора интересующего USB-устройства.

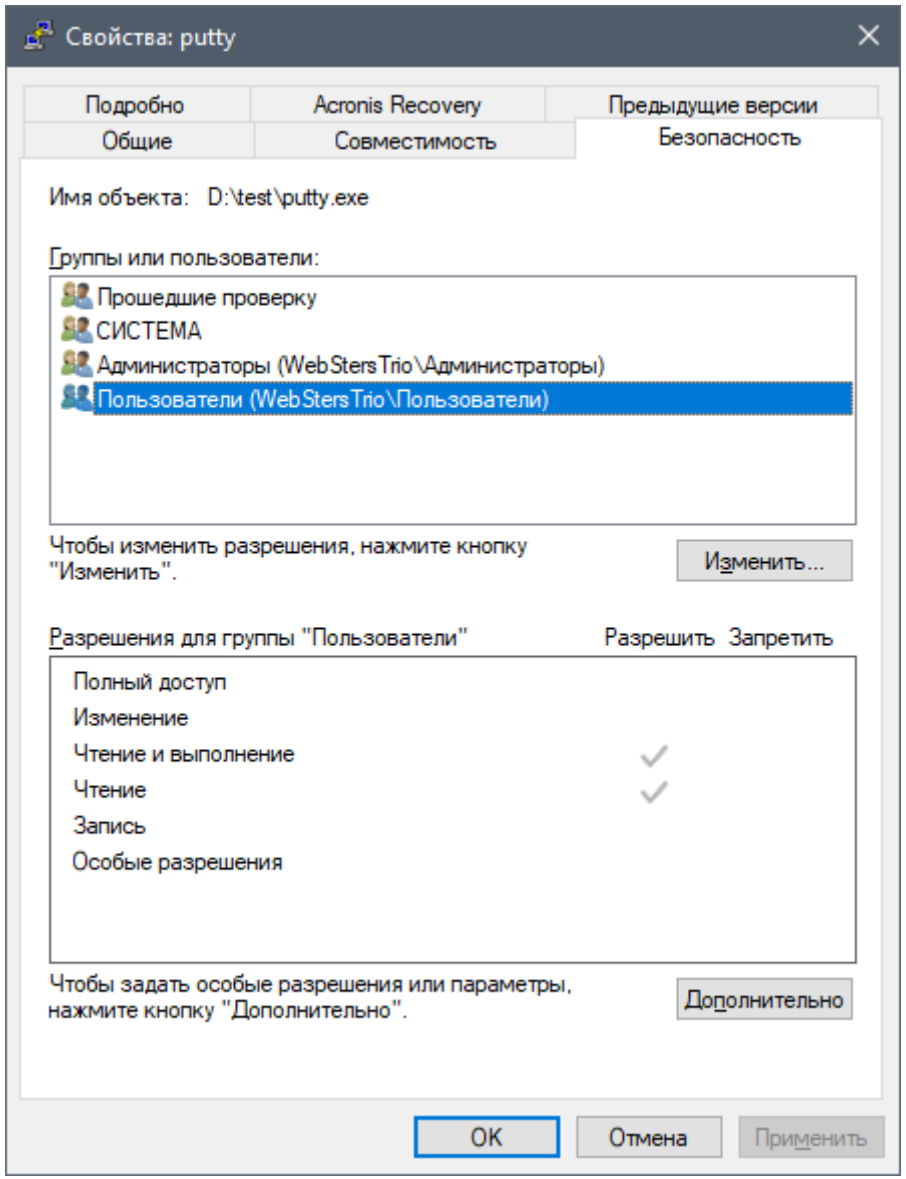
Скрипт *stop.bat* может вызываться без аргументов или с единственным ключом «-r». В последнем случае служба будет не только остановлена, но и удалена.

ИСПЫТАНИЕ УТИЛИТЫ

USB Locker


test
Средства работы с приложени...
Файл
Главная
Поделиться
Вид
Управление
Закрепить на панели быстрого доступа
Копировать
Вставить
Буфер обмена
Упорядочить
Создать папку
Создать
Свойства
Открыть
Выделить
« Data (D:) › test
Поиск: test
Имя
Дата изменения
Тип
Размер
putty
14.05.2014 18:15
Приложение
484 КБ
1 элемент
Выбран 1 элемент: 484 КБ

PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Serial
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
22
Connection type:
Raw
Telnet
Rlogin
SSH
Serial
Load, save or delete a stored session
Saved Sessions
Default Settings
VBox
Load
Save
Delete
Close window on exit:
Always
Never
Only on clean exit
About
Open
Cancel


Свойства: putty
Подробно
Acronis Recovery
Предыдущие версии
Общие
Совместимость
Безопасность
Имя объекта: D:\test\putty.exe
Группы или пользователи:
Прошедшие проверку
СИСТЕМА
Администраторы (WebStersTrio\Администраторы)
Пользователи (WebStersTrio\Пользователи)
Чтобы изменить разрешения, нажмите кнопку "Изменить".
Изменить...
Разрешения для группы "Пользователи"
Разрешить
Запретить
Полный доступ
Изменение
Чтение и выполнение
Чтение
Запись
Особые разрешения
Чтобы задать особые разрешения или параметры, нажмите кнопку "Дополнительно".
Дополнительно
OK
Отмена
Применить

Администратор: Командная строка - start.bat D:\test\putty.exe

```
C:\Users\WebSter\Documents\Visual Studio 2013\Projects\USBLocker\Debug>start.bat D:\test\putty.exe
Installing USB Locker Service...
USBLockerService is installed.
Список подключенных USB-устройств:

1) Устройство с кодом контейнера {3369C49B-807C-11E7-898A-806E6F6E696}, известное системе как:
     > HID-совместимый системный контроллер
     > Logitech Unifying USB receiver
     > Составное USB устройство
     > HID-совместимое устройство, определенное поставщиком
     > HID-совместимое устройство, определенное поставщиком
     > HID-совместимая мышь
     > Клавиатура HID
     > HID-совместимое устройство, определенное поставщиком
     > USB Input Device (Logitech Download Assistant)
     > HID-совместимое устройство управления
     > Logitech HID-compliant Unifying device
     > USB-устройство ввода

2) Устройство с кодом контейнера {B519715E-2B25-5EB2-A7B6-D9F63EBA380}, известное системе как:
     > Запоминающее устройство для USB
     > Kingston DT 101 II USB Device (Дисковый накопитель)
     > KINGSTON (DT 101 II       )
     > Том

Введите номер целевого устройства (или q для выхода):
 >
```

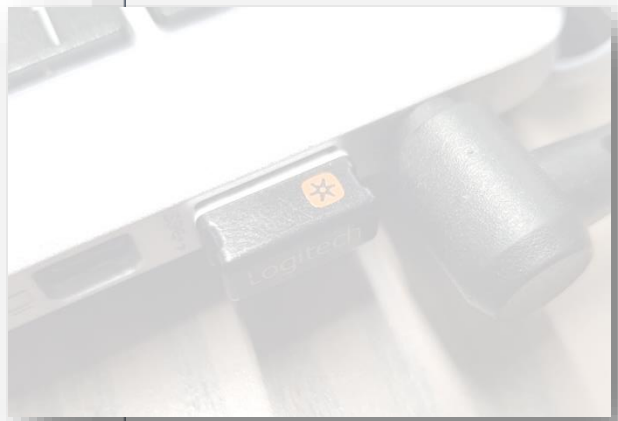
Администратор: Командная строка

```
USBLockerService is installed.
Список подключенных USB-устройств:

1) Устройство с кодом контейнера {3369C49B-807C-11E7-898A-806E6F6E696}, известное системе как:
     > HID-совместимый системный контроллер
     > Logitech Unifying USB receiver
     > Составное USB устройство
     > HID-совместимое устройство, определенное поставщиком
     > HID-совместимое устройство, определенное поставщиком
     > HID-совместимая мышь
     > Клавиатура HID
     > HID-совместимое устройство, определенное поставщиком
     > USB Input Device (Logitech Download Assistant)
     > HID-совместимое устройство управления
     > Logitech HID-compliant Unifying device
     > USB-устройство ввода

2) Устройство с кодом контейнера {B519715E-2B25-5EB2-A7B6-D9F63EBA380}, известное системе как:
     > Запоминающее устройство для USB
     > Kingston DT 101 II USB Device (Дисковый накопитель)
     > KINGSTON (DT 101 II       )
     > Том

Введите номер целевого устройства (или q для выхода):
 > 2
Выбранный код контейнера: {B519715E-2B25-5EB2-A7B6-D9F63EBA380}
Устройство успешно выбрано. Идентификатор передан службе USB Locker Service.
C:\Users\WebSter\Documents\Visual Studio 2013\Projects\USBLocker\Debug>
```



Управление компьютером

Файл Действие Вид Справка

Управление компьютером (локальн
Служебные программы
Планировщик заданий
Просмотр событий
Настраиваемые представл
Журналы Windows
Приложение
Безопасность
Установка
Система
Перенаправленные со

Уровень	Дата и время	Источник	Код события	К
Сведения	30.11.2017 5:59:27	USBLockerService	0	О
Сведения	30.11.2017 5:58:28	USBLockerService	0	О
Сведения	30.11.2017 5:56:20	USBLockerService	0	О
Сведения	30.11.2017 5:56:13	USBLockerService	0	О
Сведения	30.11.2017 5:46:00	USBLockerService	0	О
Сведения	30.11.2017 5:42:14	RestartManager	10001	О

Событие 0, USBLockerService

Общие Подробности

Режим XML

USBLockerService
Device was selected. Container ID: {B519715E-2B25-5EB2-A7B6-D9F63EBA380}

Действия

Приложение
- Открыть сохраненный журнал...
- Создать настраиваемое представ...
- Импорт настраиваемого предста...
- Очистить журнал...
- Фильтр текущего журнала...
- Свойства
- Найти...
- Сохранить все события как...
- Привязать задачу к журналу...
- Вид
- Обновить
- Справка

Событие 0, USBLockerService
- Свойства событий
- Привязать задачу к событию...
- Копировать
- Сохранить выбранные события...
- Обновить
- Справка

Дополнительные параметры безопасности для "putty"

Имя: D:\test\putty.exe

Владелец: Администраторы (WebStersTrio\Администраторы) Изменить

Разрешения | Аудит | Действующие права доступа

Для получения дополнительных сведений дважды щелкните запись разрешения. Чтобы изменить запись разрешения, выделите ее и нажмите кнопку "Изменить" (если она доступна).

Элементы разрешений:

Тип	Субъект	Доступ	Унаследовано от
Разр...	Администраторы (WebStersTrio\Админис...	Полный доступ	D:\
Разр...	СИСТЕМА	Полный доступ	D:\
Разр...	Прошедшие проверку	Изменение	D:\
Разр...	Пользователи (WebStersTrio\Пользовател...	Чтение и выполнение	D:\

Добавить Удалить Просмотреть

Отключение наследования

OK Отмена Применить

Управление компьютером

Файл Действие Вид Справка

Управление компьютером (локальн
Служебные программы

Уровень	Дата и время	Источник	Код события	К
Сведения	30.11.2017 6:05:05	USBLockerService	0	О
Сведения	30.11.2017 5:59:27	USBLockerService	0	О
Сведения	30.11.2017 5:58:28	USBLockerService	0	О
Сведения	30.11.2017 5:56:20	USBLockerService	0	О
Сведения	30.11.2017 5:56:13	USBLockerService	0	О

Событие 0, USBLockerService

Общие Подробности

Понятное представление Режим XML

+ **System**

- **EventData**

USBLockerService
File locked!

Действия

Приложение

- Открыть сохраненный журнал...
- Создать настраиваемое представ...
- Импорт настраиваемого предста...
- Очистить журнал...
- Фильтр текущего журнала...
- Свойства
- Найти...
- Сохранить все события как...
- Привязать задачу к журналу...
- Вид
- Обновить
- Справка

Событие 0, USBLockerService

- Свойства событий
- Привязать задачу к событию...
- Копировать
- Сохранить выбранные события...

Свойства: putty

Подробно | Acronis Recovery | Предыдущие версии
Общие | Совместимость | Безопасность

Имя объекта: D:\test\putty.exe

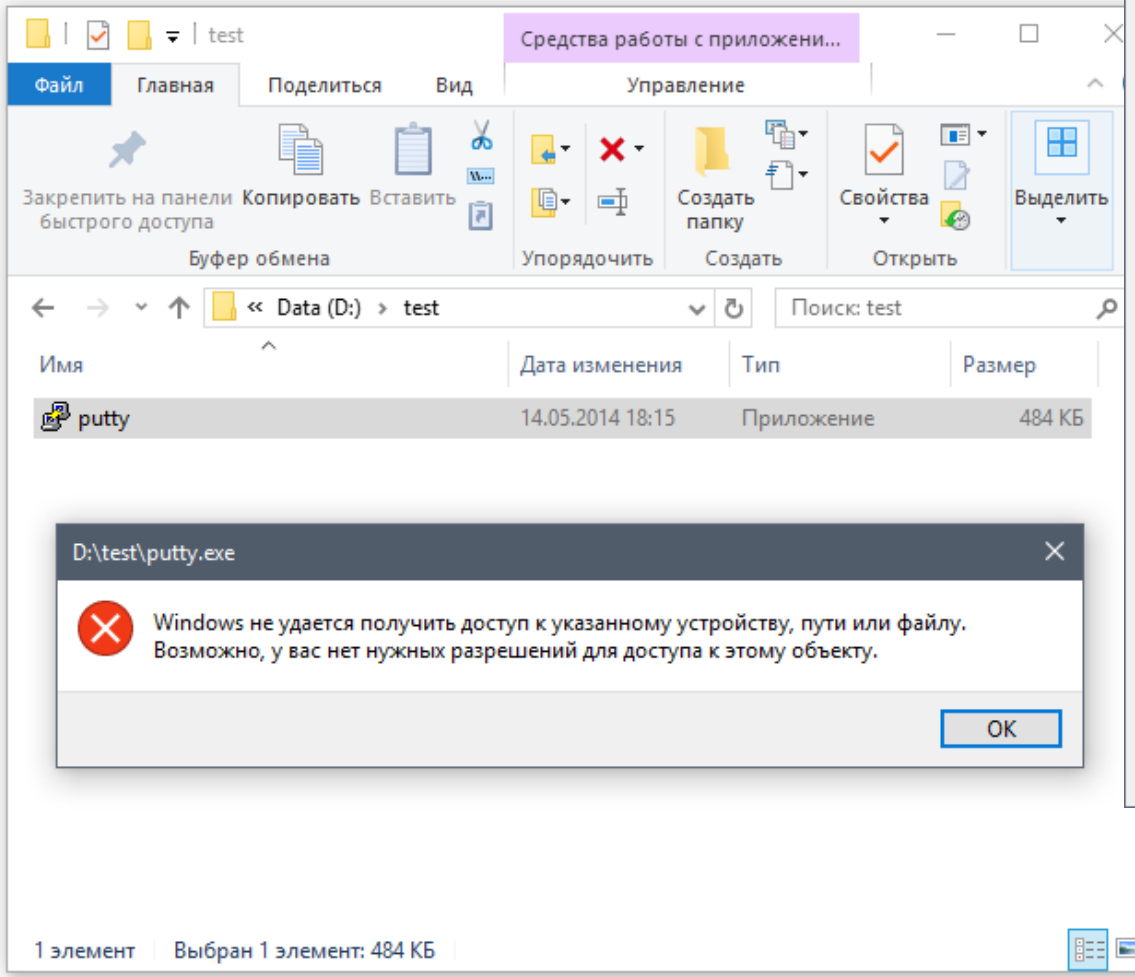
Для просмотра свойств этого объекта у вас должны быть разрешения на чтение.

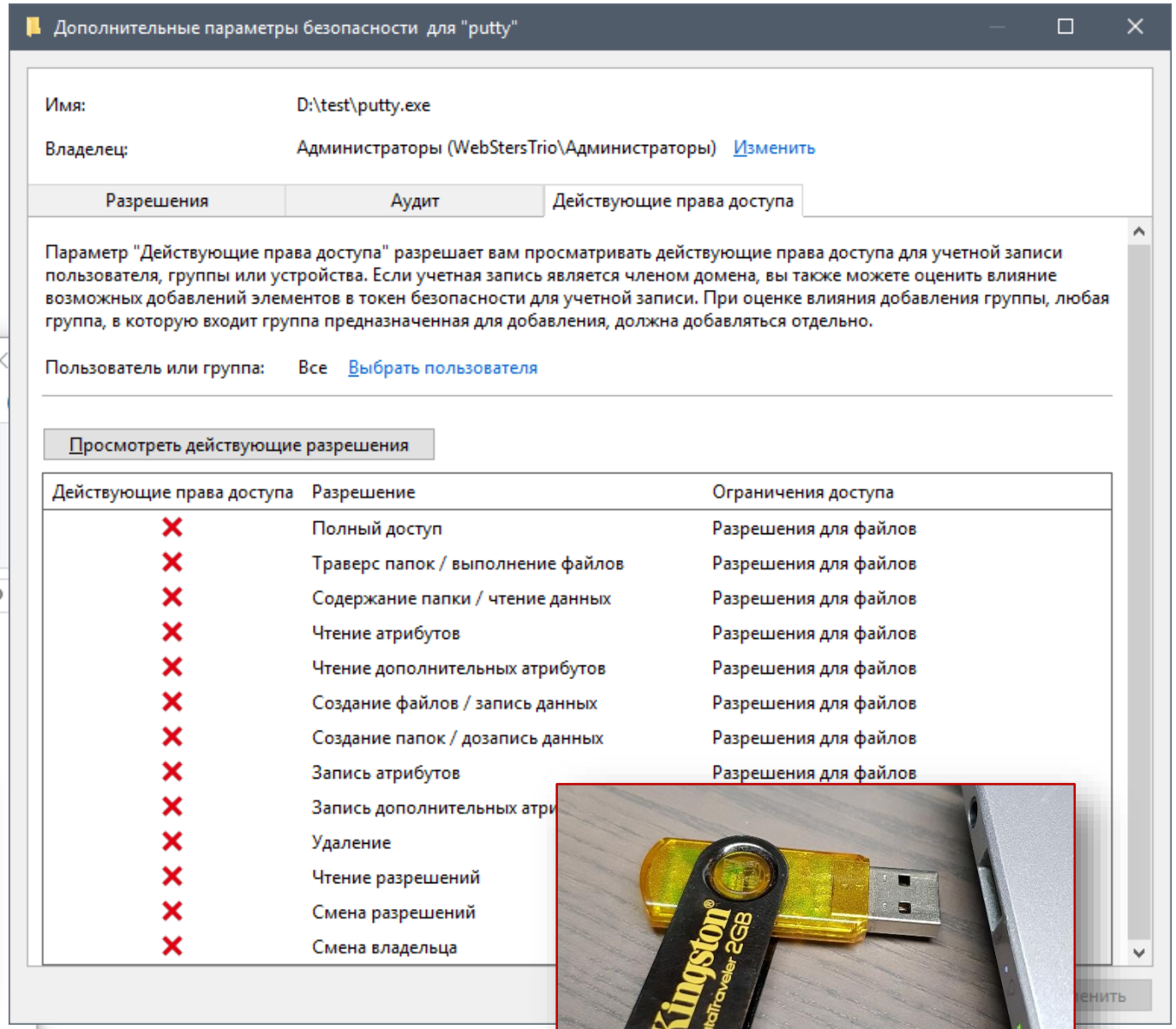
Чтобы продолжить, выберите ссылку "Дополнительно".

Чтобы задать особые разрешения или параметры, нажмите кнопку "Дополнительно".

Дополнительно

OK Отмена Применить


test
Средства работы с приложени...
Файл
Главная
Поделиться
Вид
Управление
Закрепить на панели быстрого доступа
Копировать
Вставить
Буфер обмена
Упорядочить
Создать папку
Создать
Свойства
Открыть
Выделить
« Data (D:) › test
Поиск: test
Имя
Дата изменения
Тип
Размер
putty
14.05.2014 18:15
Приложение
484 КБ
D:\test\putty.exe
Windows не удается получить доступ к указанному устройству, пути или файлу.
Возможно, у вас нет нужных разрешений для доступа к этому объекту.
OK
1 элемент
Выбран 1 элемент: 484 КБ


Дополнительные параметры безопасности для "putty"
Имя:
D:\test\putty.exe
Владелец:
Администраторы (WebStersTrio\Администраторы)
Изменить
Разрешения
Аудит
Действующие права доступа
Параметр "Действующие права доступа" разрешает вам просматривать действующие права доступа для учетной записи пользователя, группы или устройства. Если учетная запись является членом домена, вы также можете оценить влияние возможных добавлений элементов в токен безопасности для учетной записи. При оценке влияния добавления группы, любая группа, в которую входит группа предназначенная для добавления, должна добавляться отдельно.
Пользователь или группа:
Все
Выбрать пользователя
Просмотреть действующие разрешения
Действующие права доступа
Разрешение
Ограничения доступа
Полный доступ
Разрешения для файлов
Траверс папок / выполнение файлов
Разрешения для файлов
Содержание папки / чтение данных
Разрешения для файлов
Чтение атрибутов
Разрешения для файлов
Чтение дополнительных атрибутов
Разрешения для файлов
Создание файлов / запись данных
Разрешения для файлов
Создание папок / дозапись данных
Разрешения для файлов
Запись атрибутов
Разрешения для файлов
Удаление
Чтение разрешений
Смена разрешений
Смена владельца

Kingston
DataTraveler 2GB

Управление компьютером

Файл Действие Вид Справка

Управление компьютером (локальн
Служебные программы

Уровень	Дата и время	Источник	Код события	К
Сведения	30.11.2017 6:12:26	USBLockerService	0	О
Ошибка	30.11.2017 6:06:32	Application Hang	1002	(1
Сведения	30.11.2017 6:06:32	Windows Error Repo...	1001	О
Ошибка	30.11.2017 6:06:29	Apps	2486	(2
Сведения	30.11.2017 6:05:05	USBLockerService	0	О

Событие 0, USBLockerService

Общие Подробности

Понятное представление Режим XML

+ **System**

- **EventData**

USBLockerService
File unlocked!

Действия

Приложение
- Открыть сохраненный журнал...
- Создать настраиваемое представ...
- Импорт настраиваемого предста...
- Очистить журнал...
- Фильтр текущего журнала...
- Свойства
- Найти...
- Сохранить все события как...
- Привязать задачу к журналу...
- Вид
- Обновить
- Справка

Событие 0, USBLockerService
- Свойства событий
- Привязать задачу к событию...
- Копировать
- Сохранить выбранные события...
- Обновить
- Справка

Свойства: putty

Подробно | Acronis Recovery | Предыдущие версии
Общие | Совместимость | Безопасность

Имя объекта: D:\test\putty.exe

Группы или пользователи:

- Все
- Прошедшие проверку
- СИСТЕМА
- Администраторы (WebStersTrio\Администраторы)
- Пользователи (WebStersTrio\Пользователи)

Чтобы изменить разрешения, нажмите кнопку "Изменить".

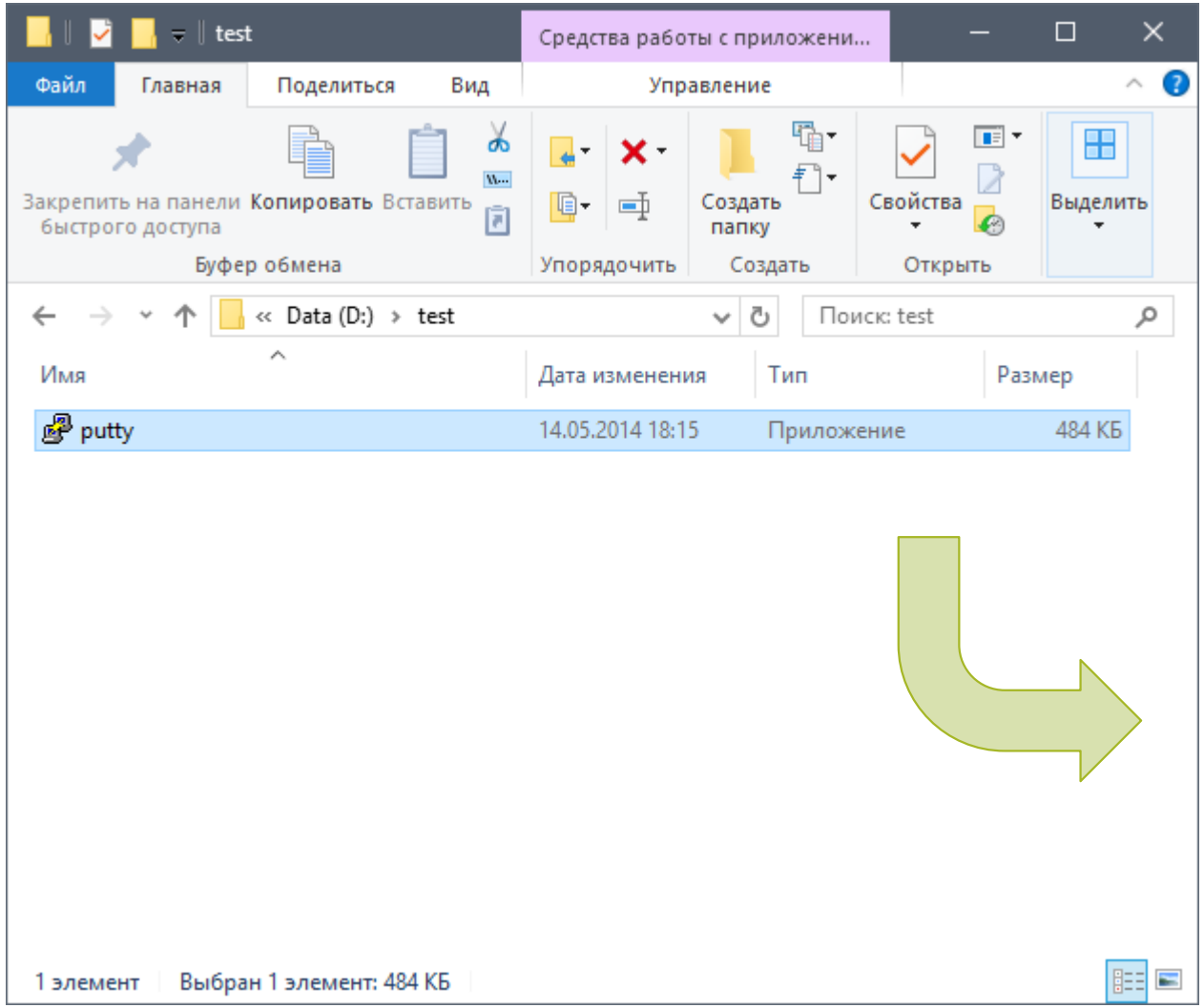
Изменить...

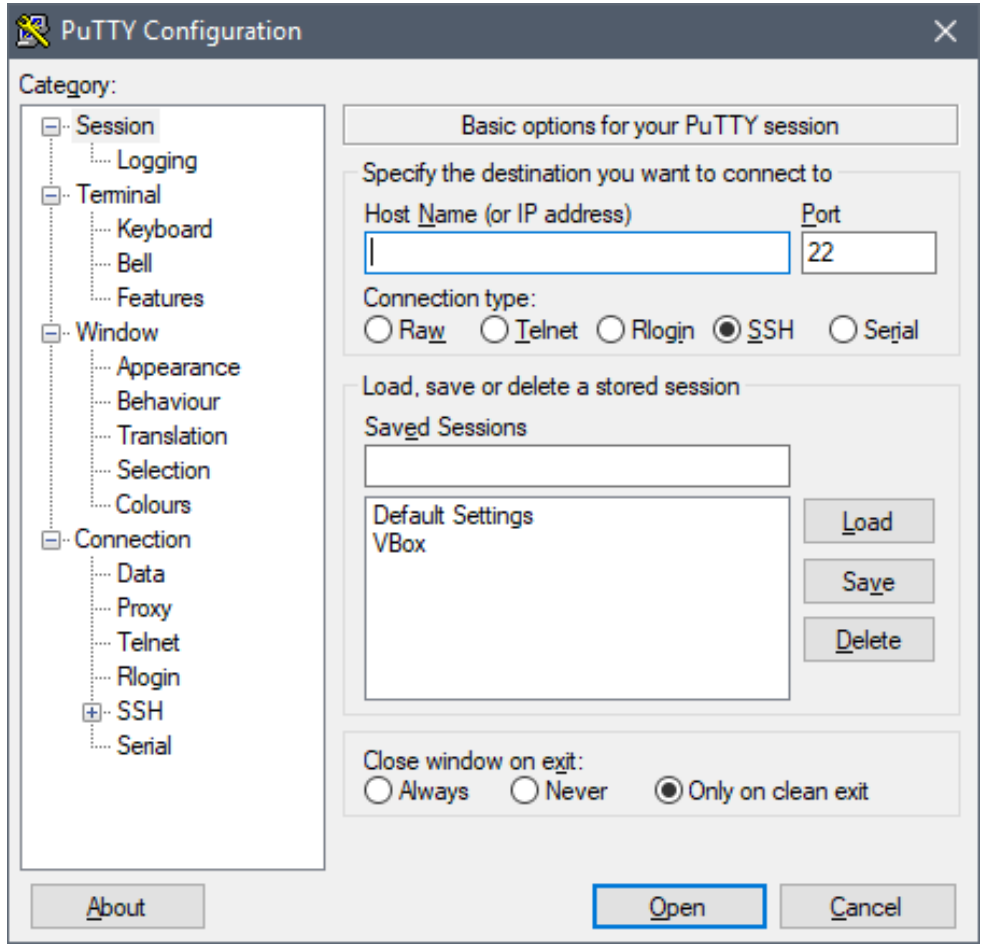
Разрешения для группы "Все"	Разрешить	Запретить
Полный доступ		
Изменение		
Чтение и выполнение	✓	
Чтение	✓	
Запись		
Особые разрешения		

Чтобы задать особые разрешения или параметры, нажмите кнопку "Дополнительно".

Дополнительно

OK Отмена Применить


test
Средства работы с приложени...
Файл
Главная
Поделиться
Вид
Управление
Закрепить на панели быстрого доступа
Копировать
Вставить
Буфер обмена
Создать папку
Упорядочить
Создать
Свойства
Открыть
Выделить
« Data (D:) › test
Поиск: test
Имя
Дата изменения
Тип
Размер
putty
14.05.2014 18:15
Приложение
484 КБ
1 элемент
Выбран 1 элемент: 484 КБ


PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Serial
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
22
Connection type:
Raw
Telnet
Rlogin
SSH
Serial
Load, save or delete a stored session
Saved Sessions
Default Settings
VBox
Load
Save
Delete
Close window on exit:
Always
Never
Only on clean exit
About
Open
Cancel

СПАСИБО ЗА ВНИМАНИЕ