Санкт-Петербургский политехнический университет Петра Великого

Системная изоляция процессов в Windows средствами LPAC

Курс: **Проектирование ОС и компонентов**

Студент: **Д.В. Круминьш**
Группа: **13541/3**

Преподаватель: **Е.В. Душутина**

ПОЛИТЕХ

ТЕОРИЯ

Причины потребности в изоляции

Современные серверы обладают избыточной производительностью, и приложения порой не используют даже их части.

Выходом стала **виртуализация**, позволяющая запускать несколько ОС на одном сервере, гарантированно разделяя их между собой и выделяя каждой нужное количество ресурсов.

Следующий этап — **микросервисы**, когда каждая часть приложения развертывается отдельно, как самодостаточный, изолированный компонент.

В ОС семейства **Windows** данным инструментом служит - **Application Container**.

Application Container

Application Container - это иной тип виртуализации, предоставляющий обособленную среду для выполнения приложений, называемую OS Virtualization. Реализуются контейнеры за счет использования изолированного пространства имен, включающего все необходимые для работы ресурсы (виртуализированные имена), с которыми можно взаимодействовать (файлы, сетевые порты, процессы и прочее) и выйти за которые нельзя.

Контейнеры используют одно и то же **ядро ОС**. В отличие от виртуальных машин, контейнеры не полагаются на отдельный **слой гипервизора**.

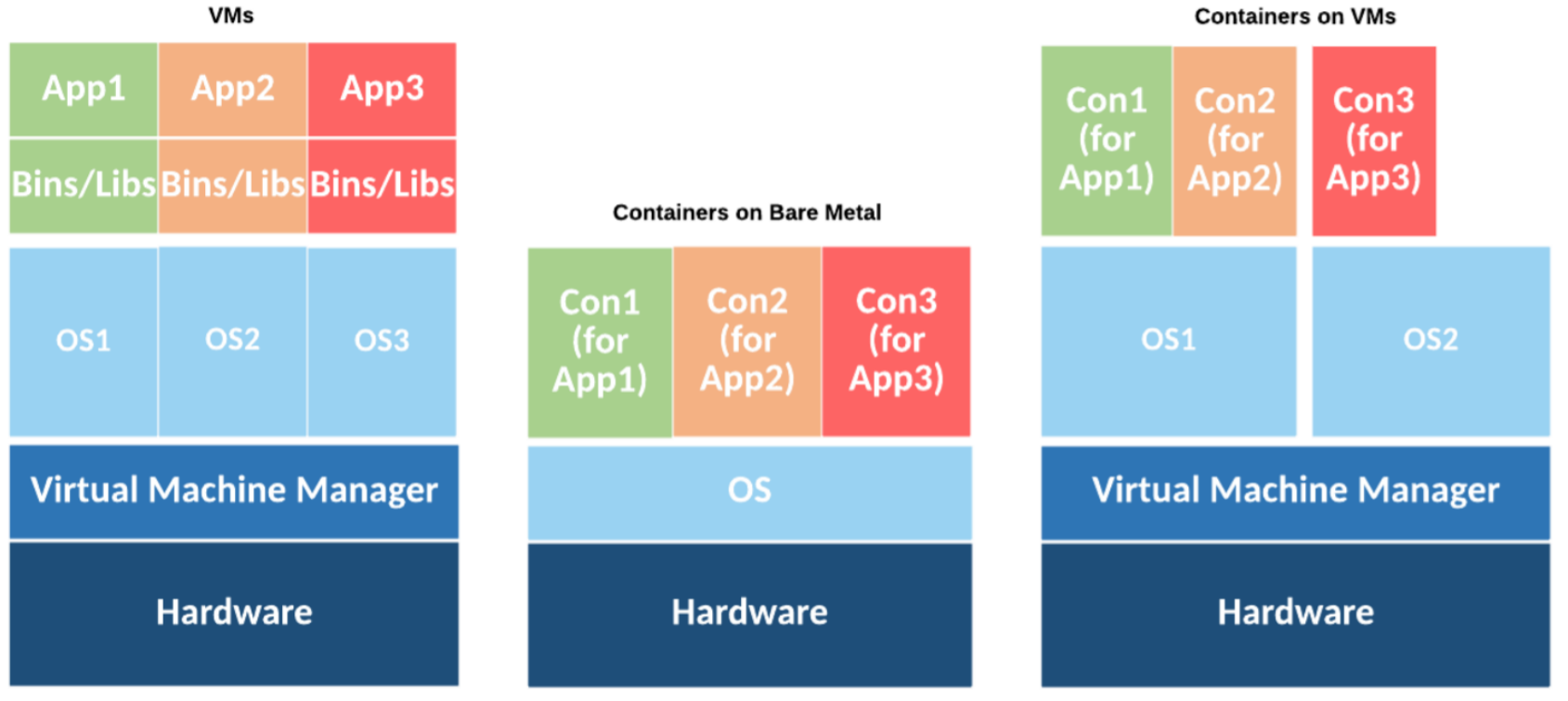
Приложение внутри контейнера считает, что оно единственное, и работает в полноценной ОС без каких-либо ограничений.

Упрощенный пример контейнеризации

Кухня
(контейнер)

Дом
(Операционная система)

Реализация виртуальной машины и контейнера



Сравнение реализаций виртуальной машины и контейнера

Типы контейнеров

LPAC является частным случаем Application Container.

- **AC** - Application Container
  - Появился в Windows Server 2016;
  - Разработчики были вдохновлены успехами Docker(Linux).
- **LPAC** - Less Privileged Application Container
  - Появился в Windows 10 Creators Update(2017 год).

Изоляция LPAC

Когда приложение запущено внутри LPAC, все разрешения ему требуется выдавать явно.

На процесс накладываются следующие ограничения доступа:

- К устройствам(микрофон, камера, принтер...);
- Файлам системы(чтение, запись);
- Процессам(межпроцессные взаимодействия);
- Сети(использование сетевых интерфейсов).

ПРАКТИКА

Порядок запуска

Для запуска LPAC контейнера необходимо:

1. Создать контейнер, с помощью функции **CreateAppContainerProfile**;
2. [Опционально] Установка общих доступов, с помощью функции **CreateWellKnownSid**;
3. [Опционально] Установка конкретных доступов, например для файлов используя функцию **SetEntriesInAclA**;
4. [Опционально] Изменение переменных окружения, с помощью функции **ExpandEnvironmentStringsA**;
5. Запуск программы, с помощью **CreateProcessA**.

Создание контейнера

```
1  HRESULT WINAPI CreateAppContainerProfile(
2    _In_  PCWSTR                pszAppContainerName,
3    _In_  PCWSTR                pszDisplayName,
4    _In_  PCWSTR                pszDescription,
5    _In_  PSID_AND_ATTRIBUTES   pCapabilities,
6    _In_  DWORD                 dwCapabilityCount,
7    _Out_ PSID                  *ppSidAppContainerSid
8  );
```

Листинг 1: Прототип CreateAppContainerProfile

```
1  HRESULT WINAPI DeriveAppContainerSidFromAppContainerName(
2    _In_  PCWSTR pszAppContainerName,
3    _Out_ PSID   *ppsidAppContainerSid
4  );
```

Листинг 2: Прототип DeriveAppContainerSidFromAppContainerName

Затем идет вызов **UpdateProcThreadAttribute**.

Security Identifier (SID)

Security Identifier (SID) — идентификатор безопасности, структура данных в Windows, которая может идентифицировать системные объекты, например элементы управления доступом (Access Control Entries, ACE), токены доступа (Access Token), дескрипторы безопасности (Security Descriptor). SID всегда начинается с буквы S, далее идут числа, которые обозначают номер редакции ОС, источники выдачи, удостоверяющие центры и другую информацию.

HKEY_CURRENT_USER/Software/Classes/Local Settings/Software/Microsoft/Windows/CurrentVersion/AppContainerStorage/Mappings

Установка общих доступов

```
BOOL WINAPI CreateWellKnownSid(
  _In_      WELL_KNOWN_SID_TYPE WellKnownSidType,
  _In_opt_  PSID                DomainSid,
  _Out_opt_ PSID                pSid,
  _Inout_   DWORD               *cbSid
);
```

Листинг 3: Прототип CreateWellKnownSid

WELL_KNOWN_SID_TYPE - идентификатор безопасности (имеется 94 идентификатора).

```
typedef struct _SECURITY_CAPABILITIES {
  SID                   AppContainerSid;
  PSID_AND_ATTRIBUTES   Capabilities;
  DWORD                 CapabilityCount;
  DWORD                 Reserved;
} SECURITY_CAPABILITIES, *PSECURITY_CAPABILITIES;
```

Листинг 4: Структура SECURITY_CAPABILITIES

Установка общих доступов

```
1  BOOL WINAPI UpdateProcThreadAttribute(
2    _Inout_   LPPROC_THREAD_ATTRIBUTE_LIST lpAttributeList,
3    _In_      DWORD                        dwFlags,
4    _In_      DWORD_PTR                    Attribute,
5    _In_      PVOID                        lpValue,
6    _In_      SIZE_T                       cbSize,
7    _Out_opt_ PVOID                        lpPreviousValue,
8    _In_opt_  PSIZE_T                      lpReturnSize
9  );
```

Листинг 5: Прототип UpdateProcThreadAttribute

- **Attribute** - PROC_THREAD_ATTRIBUTE_SECURITY_CAPABILITIES - означает, что следующий, созданный процесс будет создан в контейнере;
- **lpValue** - структура SECURITY_CAPABILITIES.

Изменение переменных окружения

```
1 BOOL WINAPI SetEnvironmentVariable(
2   _In_     LPCTSTR lpName,
3   _In_opt_ LPCTSTR lpValue
4 );
```

Листинг 6: Прототип SetEnvironmentVariable

```
1 DWORD WINAPI ExpandEnvironmentStrings(
2   _In_      LPCTSTR lpSrc,
3   _Out_opt_ LPTSTR  lpDst,
4   _In_      DWORD   nSize
5 );
```

Листинг 7: Прототип ExpandEnvironmentStrings

- **SetEnvironmentVariable** - установка/перезапись переменной;
- **ExpandEnvironmentStrings** - получение значения переменной.

Установка конкретных доступов к файлам

```
1 typedef struct _EXPLICIT_ACCESS {
2   DWORD       grfAccessPermissions;
3   ACCESS_MODE grfAccessMode;
4   DWORD       grfInheritance;
5   TRUSTEE     Trustee;
6 } EXPLICIT_ACCESS, *PEXPLICIT_ACCESS;
```

Листинг 8: Структура EXPLICIT_ACCESS

```
1 typedef struct _TRUSTEE {
2   PTRUSTEE                  pMultipleTrustee;
3   MULTIPLE_TRUSTEE_OPERATION MultipleTrusteeOperation;
4   TRUSTEE_FORM              TrusteeForm;
5   TRUSTEE_TYPE              TrusteeType;
6   LPCH                      ptstrName;
7 } TRUSTEE, *PTRUSTEE;
```

Листинг 9: Структура TRUSTEE

Установка конкретных доступов к файлам

Получение дескриптора безопасности

```
1  DWORD WINAPI GetNamedSecurityInfo(
2    _In_      LPTSTR                pObjectName,
3    _In_      SE_OBJECT_TYPE        ObjectType,
4    _In_      SECURITY_INFORMATION  SecurityInfo,
5    _Out_opt_ PSID                  *ppsidOwner,
6    _Out_opt_ PSID                  *ppsidGroup,
7    _Out_opt_ PACL                  *ppDacl,
8    _Out_opt_ PACL                  *ppSacl,
9    _Out_opt_ PSECURITY_DESCRIPTOR *ppSecurityDescriptor
10 );
```

Листинг 10: Прототип GetNamedSecurityInfo

В возращаемом указателе **ppDacl** права для объекта.

Установка конкретных доступов к файлам

```
1 DWORD WINAPI SetEntriesInAcl(
2   _In_     ULONG            cCountOfExplicitEntries,
3   _In_opt_ PEXPLICIT_ACCESS pListOfExplicitEntries,
4   _In_opt_ PACL             OldAcl,
5   _Out_    PACL             *NewAcl
6 );
```

Листинг 11: Структура SetEntriesInAcl

Установка конкретных доступов к файлам

Установка обновленного дескриптора через **pDacl**.

```
1  DWORD WINAPI SetNamedSecurityInfo(
2    _In_      LPTSTR                pObjectName,
3    _In_      SE_OBJECT_TYPE        ObjectType,
4    _In_      SECURITY_INFORMATION  SecurityInfo,
5    _In_opt_  PSID                  psidOwner,
6    _In_opt_  PSID                  psidGroup,
7    _In_opt_  PACL                  pDacl,
8    _In_opt_  PACL                  pSacl
9  );
```

Листинг 12: Структура SetNamedSecurityInfo

Запуск программы

```
1  BOOL WINAPI CreateProcess(
2    _In_opt_    LPCTSTR                lpApplicationName,
3    _Inout_opt_ LPTSTR                 lpCommandLine,
4    _In_opt_    LPSECURITY_ATTRIBUTES  lpProcessAttributes,
5    _In_opt_    LPSECURITY_ATTRIBUTES  lpThreadAttributes,
6    _In_        BOOL                   bInheritHandles,
7    _In_        DWORD                  dwCreationFlags,
8    _In_opt_    LPVOID                 lpEnvironment,
9    _In_opt_    LPCTSTR                lpCurrentDirectory,
10   _In_        LPSTARTUPINFO          lpStartupInfo,
11   _Out_       LPPROCESS_INFORMATION  lpProcessInformation
12 );
```

Листинг 13: Прототип CreateProcessA

ЭКСПЕРИМЕНТЫ

Способ тестирования

```
1  OpenProcessToken(GetCurrentProcess(), TOKEN_QUERY, &process_token);
2
3  if (!GetTokenInformation(process_token, TokenIsAppContainer, &
       ↪ is_container, sizeof(is_container), &return_length))
4      return false;
5  return true;
```

Листинг 14: Фрагмент кода, по проверки работы в контейнере

Если приложение в песочнице, провести эксперименты с:

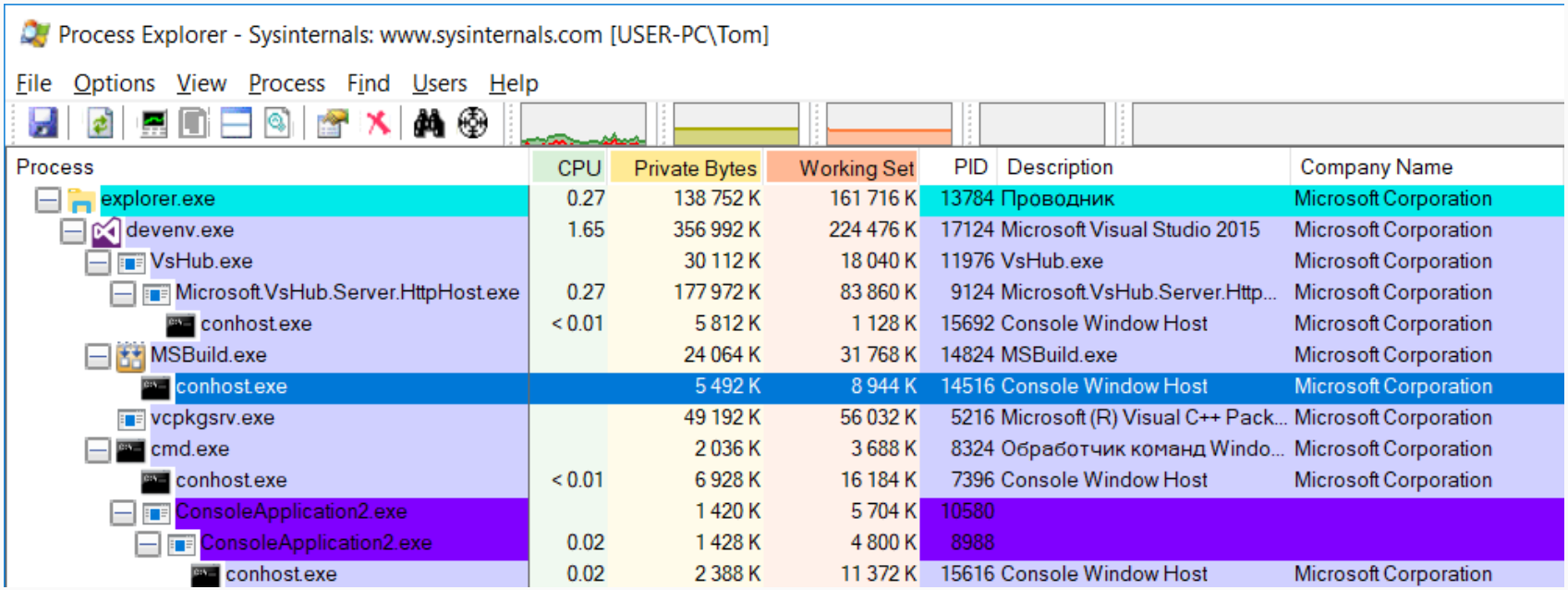
- переменными окружения;
- файлами;
- сетью;
- проверкой изоляции.

Изменение переменных окружения

```
1  SetEnvironmentVariable(L"testName", L"testValue");
2
3  SetEnvironmentVariable(L"USERDOMAIN", L"HELLOWORLD");
4
5  ExpandEnvironmentStringsA("%temp%", path, MAX_PATH - 1);
6  printf("New path of %%temp%%: %s\n", path);
7
8  ExpandEnvironmentStringsA("%localappdata%", path, MAX_PATH - 1);
9  printf("New path of %%localappdata%%: %s\n", path);
```

Листинг 15: Фрагмент кода, по изменению переменных окружения

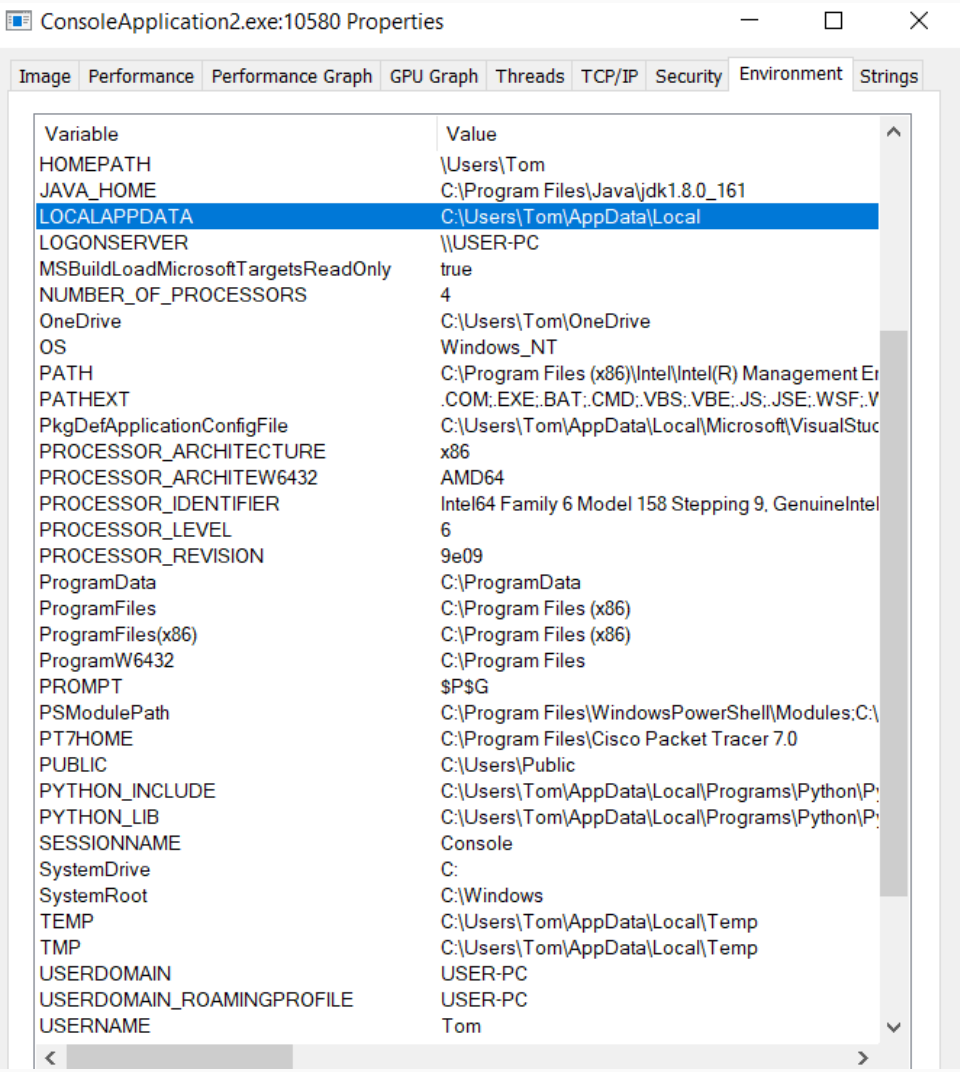
Process Explorer

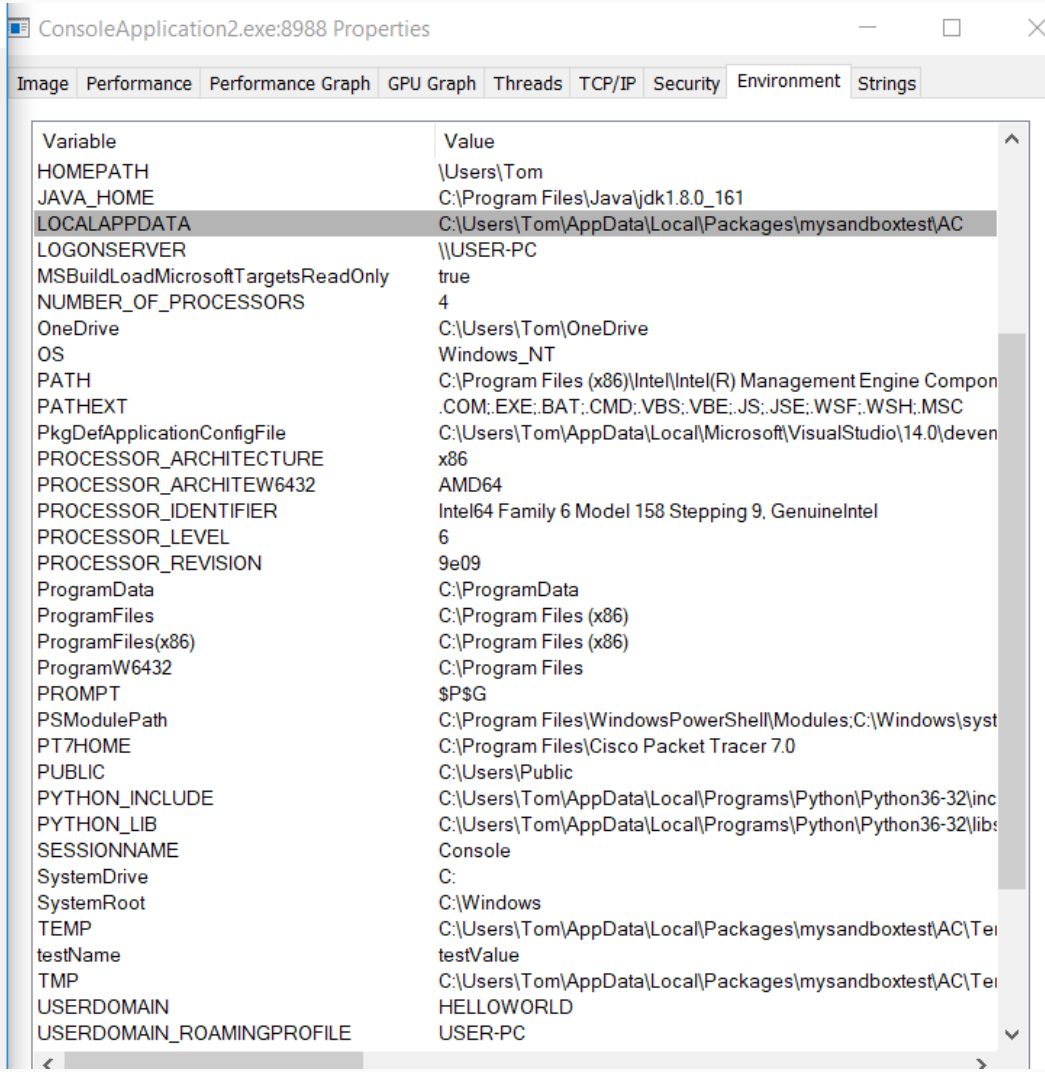


Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
explorer.exe	0.27	138 752 K	161 716 K	13784	Проводник	Microsoft Corporation
devenv.exe	1.65	356 992 K	224 476 K	17124	Microsoft Visual Studio 2015	Microsoft Corporation
VsHub.exe		30 112 K	18 040 K	11976	VsHub.exe	Microsoft Corporation
Microsoft.VsHub.Server.HttpHost.exe	0.27	177 972 K	83 860 K	9124	Microsoft.VsHub.Server.Http...	Microsoft Corporation
conhost.exe	< 0.01	5 812 K	1 128 K	15692	Console Window Host	Microsoft Corporation
MSBuild.exe		24 064 K	31 768 K	14824	MSBuild.exe	Microsoft Corporation
conhost.exe		5 492 K	8 944 K	14516	Console Window Host	Microsoft Corporation
vcpkgsrv.exe		49 192 K	56 032 K	5216	Microsoft (R) Visual C++ Pack...	Microsoft Corporation
cmd.exe		2 036 K	3 688 K	8324	Обработчик команд Windo...	Microsoft Corporation
conhost.exe	< 0.01	6 928 K	16 184 K	7396	Console Window Host	Microsoft Corporation
ConsoleApplication2.exe		1 420 K	5 704 K	10580		
ConsoleApplication2.exe	0.02	1 428 K	4 800 K	8988		
conhost.exe	0.02	2 388 K	11 372 K	15616	Console Window Host	Microsoft Corporation

Древовидная стрктура процессов

Сравнение переменных окружения





Доступ к файлам

```
1 EXPLICIT_ACCESS_A explicit_access;
2 explicit_access.grfAccessMode = GRANT_ACCESS;
3 explicit_access.grfAccessPermissions = FILE_ALL_ACCESS;
4 ...
5 GetNamedSecurityInfoA(object_name, object_type,
      ↪ DACL_SECURITY_INFORMATION, NULL, NULL, &original_acl, NULL,
      ↪ NULL);
6 SetEntriesInAclA(1, &explicit_access, original_acl, &new_acl);
7 SetNamedSecurityInfoA(object_name, object_type,
      ↪ DACL_SECURITY_INFORMATION, NULL, NULL, new_acl, NULL);
```

Листинг 16: Фрагмент кода, по доступу к файлам

```
1 Opening of file C:\Users\Tom\AppData\Local\Packages\mysandboxtest\AC\
      ↪ Temp\allowed_test.txt was successful
2 Opening of file C:\Users\Tom\desktop\allowed_test.txt was successful
3 Opening of file C:\Users\Tom\desktop\blocked_test.txt returned access
      ↪ denied
```

Листинг 17: Результат

Проверка доступ к сети

Тестирование будет производиться путем открытия 80 порта, к следующим адреса:

- **108.177.14.138** (google.com);
- **192.168.1.1** (локальный адрес роутера).

Тесты будут выполнены с использованием следующих: WELL_KNOWN_SID_TYPE:

- WinCapabilityPrivateNetworkClientServerSid - доступ к локальной сети;
- WinCapabilityInternetClientServerSid - доступ к сети ”Интернет”.

Проверка доступ к сети

Без задачи каких-либо WELL_KNOWN_SID_TYPE:

```
1 Connection to 108.177.14.138 was blocked
2 Connection to 192.168.1.1 was blocked
```

Листинг 18: Полный запрет сети

WinCapabilityPrivateNetworkClientServerSid:

```
1 Connection to 108.177.14.138 was blocked
2 Connection to 192.168.1.1 was successful
```

Листинг 19: Доступ только к локальной сети

Проверка доступ к сети

WinCapabilityInternetClientServerSid:

```
1 Connection to 108.177.14.138 was successful
2 Connection to 192.168.1.1 was blocked
```

Листинг 20: Доступ только к сети ”Интернет”

Вместе:

```
1 Connection to 108.177.14.138 was successful
2 Connection to 192.168.1.1 was successful
```

Листинг 21: Полный доступ к сети

Список процессов

```
1  Found process: [System Process]
2  Found process: ConsoleApplication2.exe
3  Found process: conhost.exe
```

Листинг 22: Список процессов из изолированного процесса

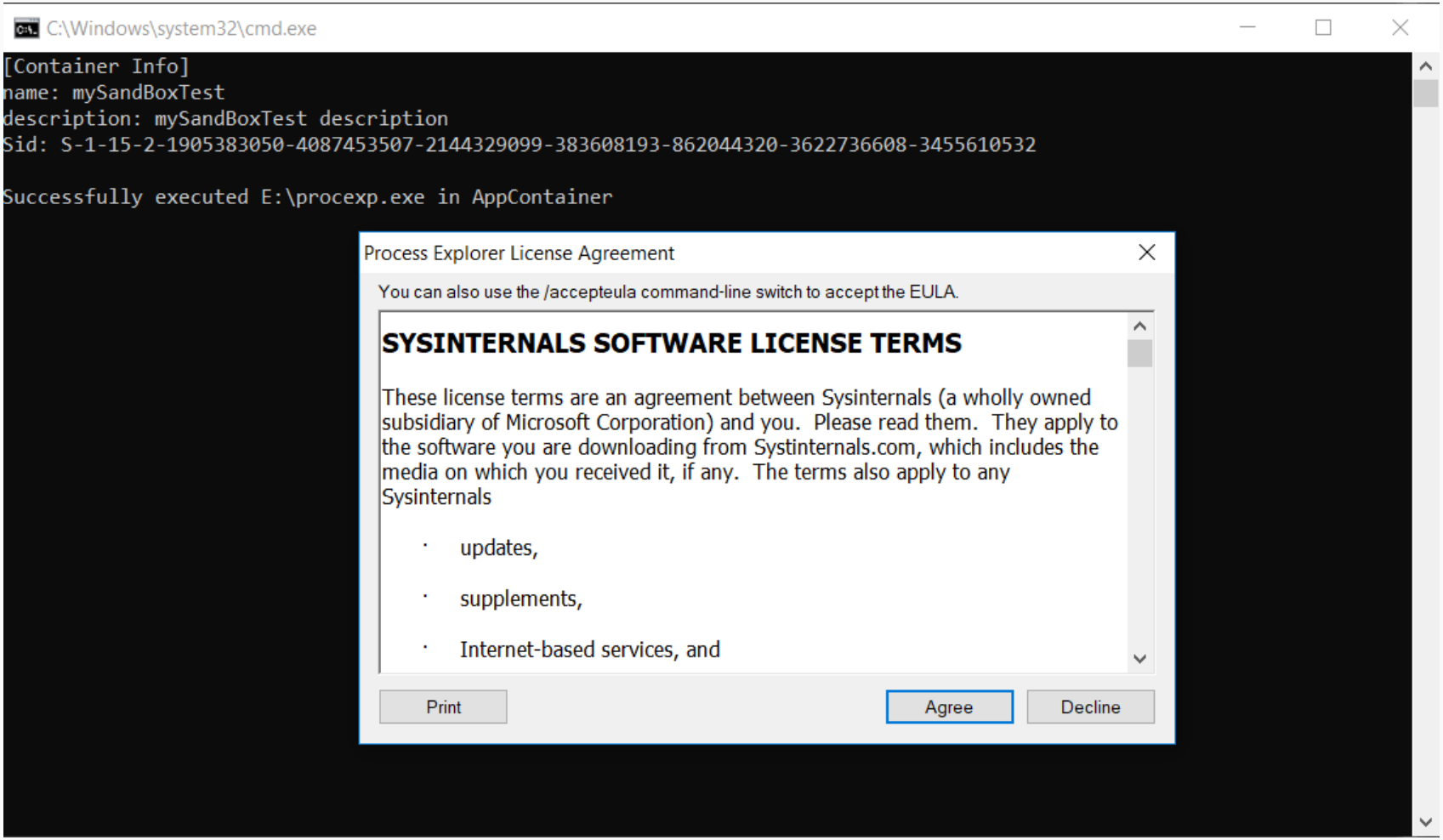
```
1  Found process: [System Process]
2  Found process: System
3  Found process: smss.exe
4  Found process: csrss.exe
5  Found process: wininit.exe
6  Found process: services.exe
7  Found process: lsass.exe
8  Found process: svchost.exe
9  Found process: WUDFHost.exe
10 Found process: fontdrvhost.exe
11 Found process: svchost.exe
12 Found process: WUDFHost.exe
13 Found process: svchost.exe
14 ...
```

Листинг 23: Список процессов из обычного процесса

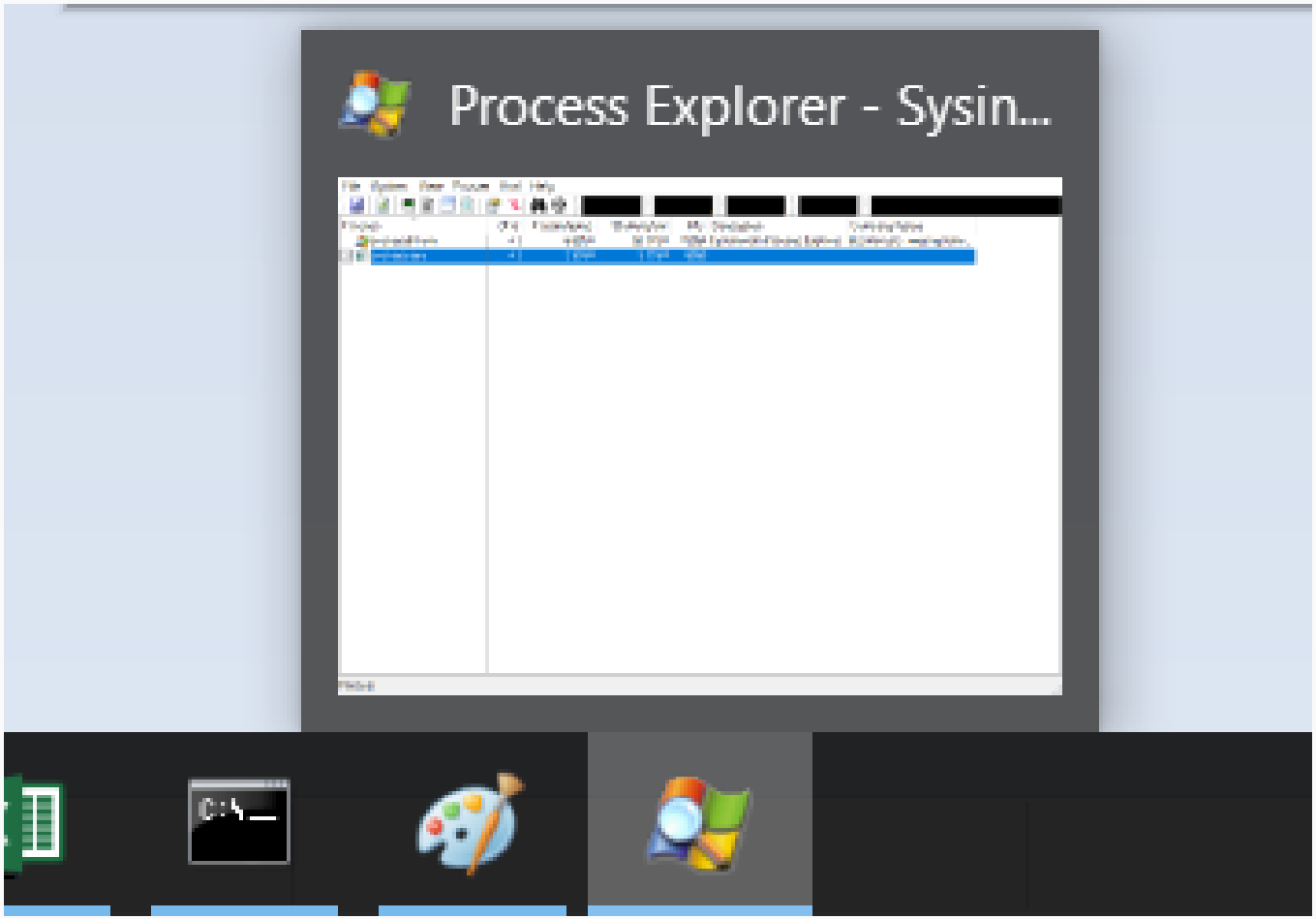
Запуск прочих программ

Название	Работоспособность
Консоль	
Блокнот	
UltraISO	
Process Explorer	
Internet Explorer	
Google Chrome	

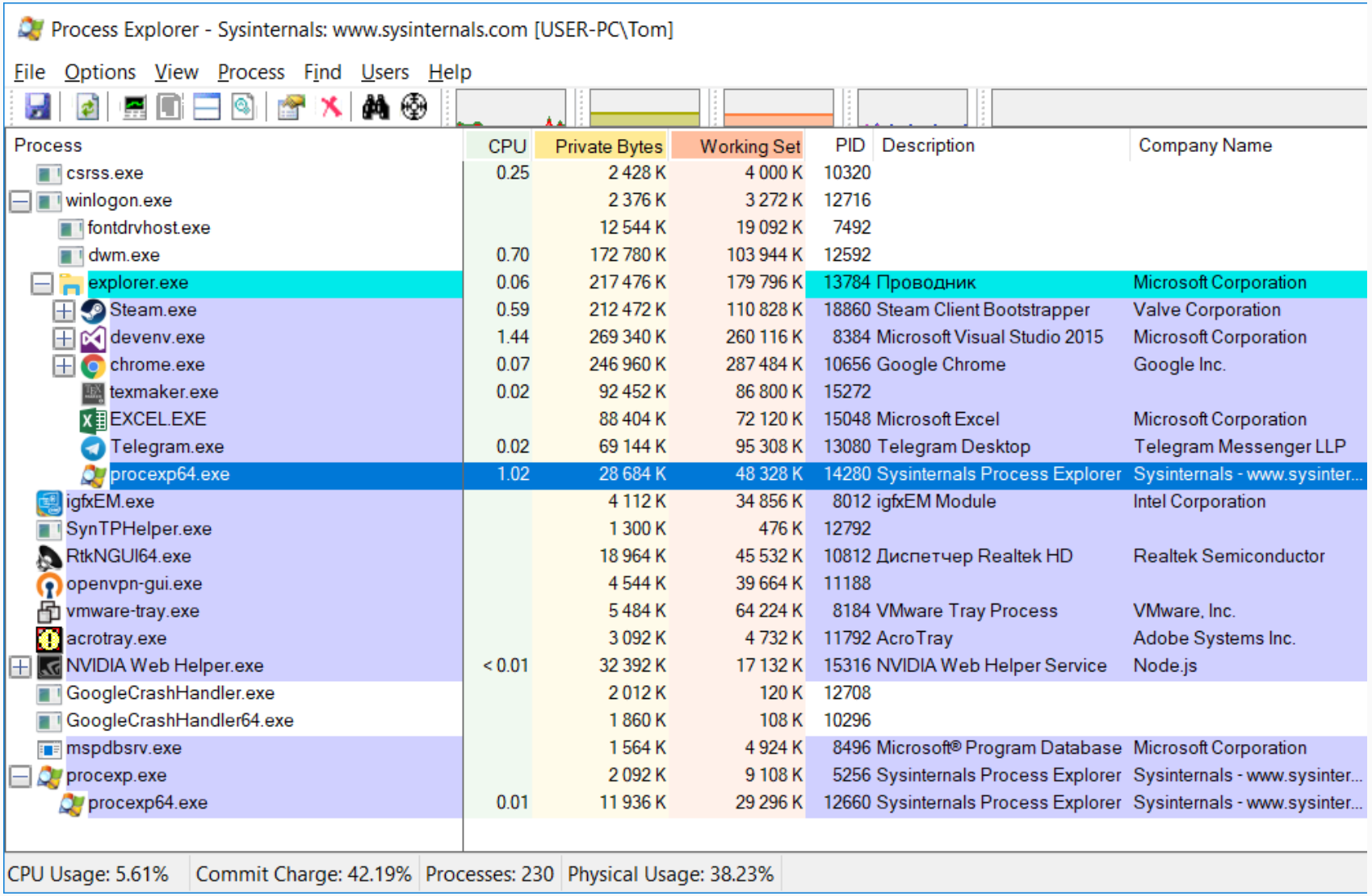
Запуск Process Explorer



Запуск Process Explorer

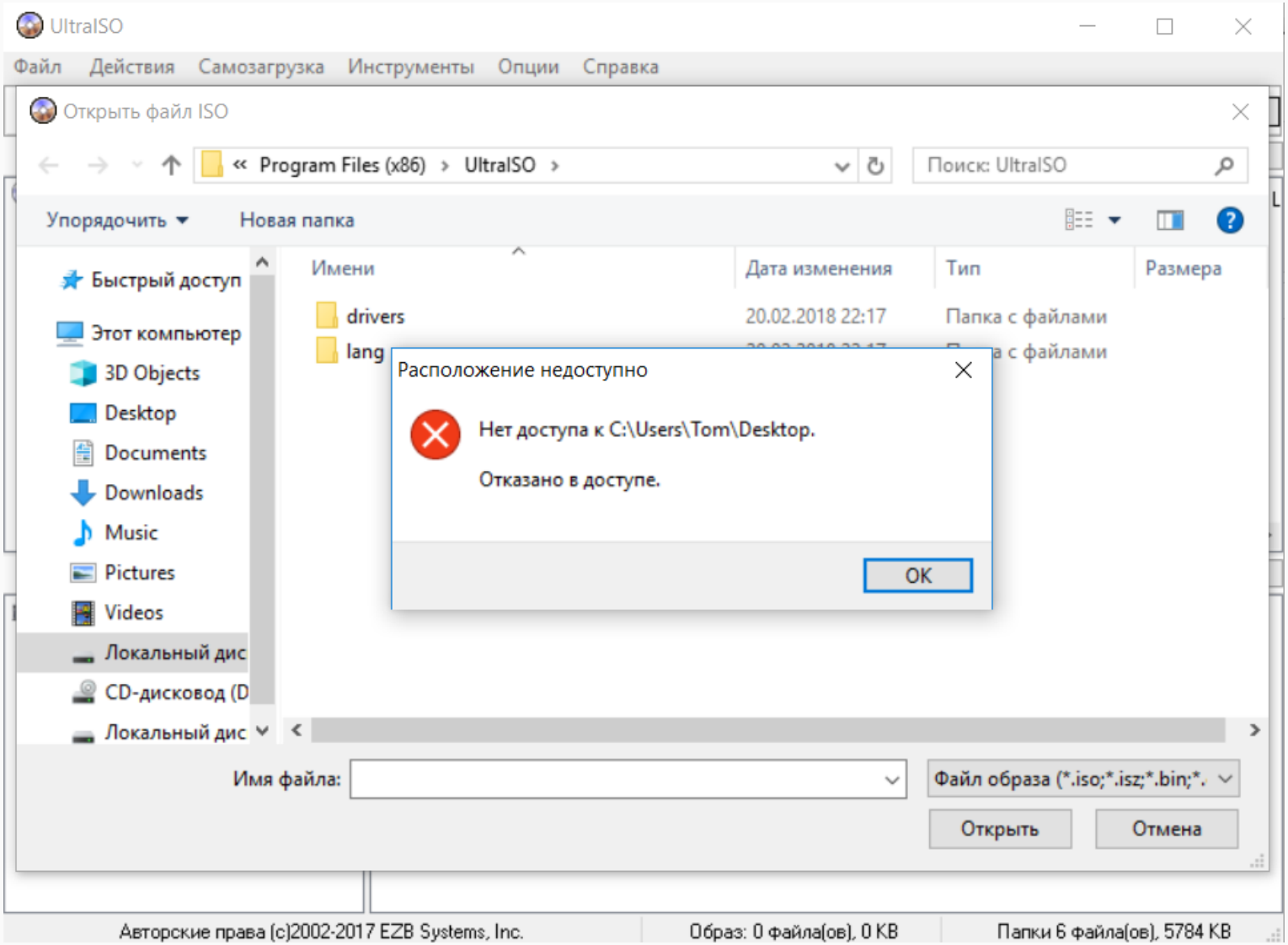


Реакция системы

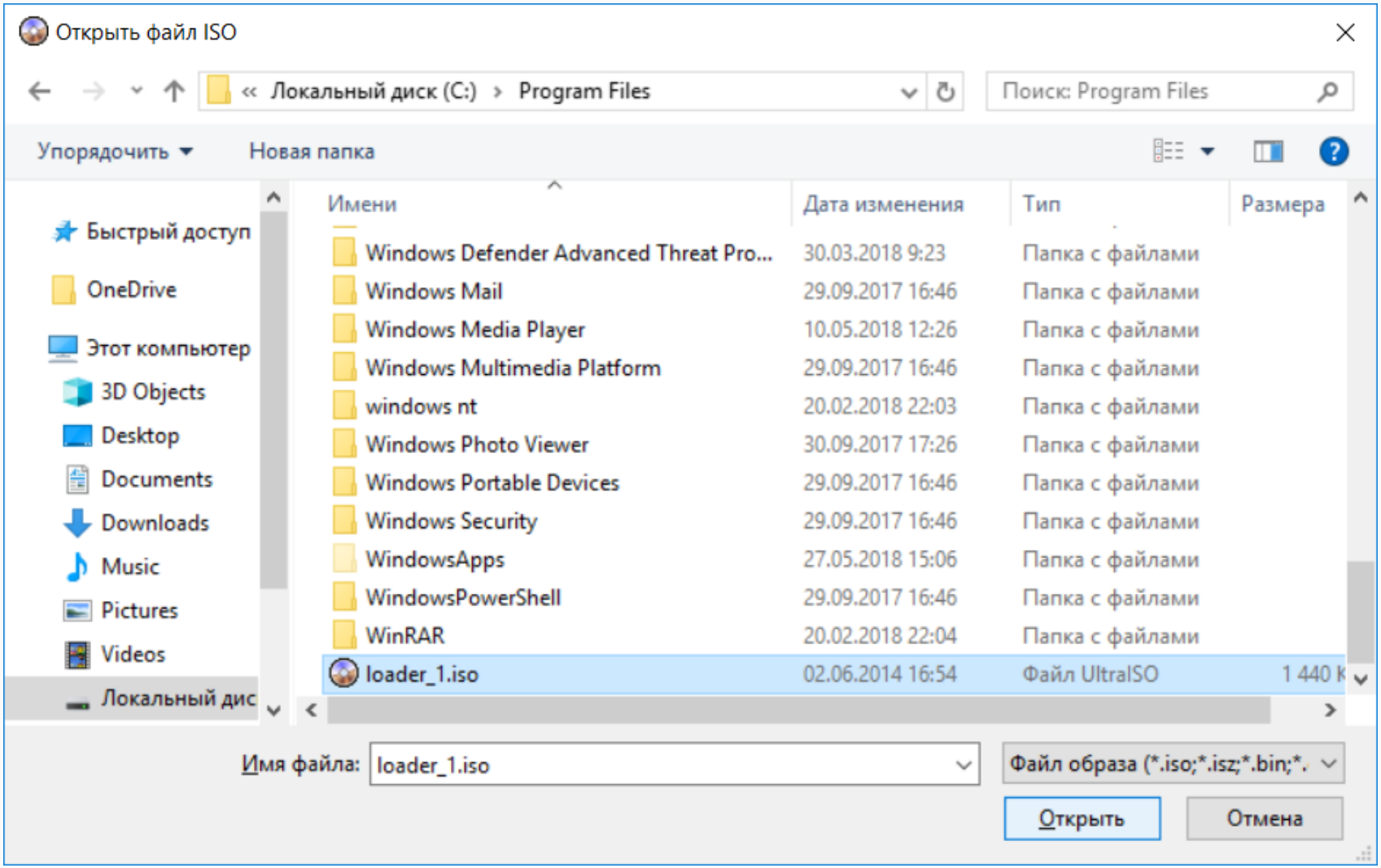


Состояние после закрытия родительского процесса

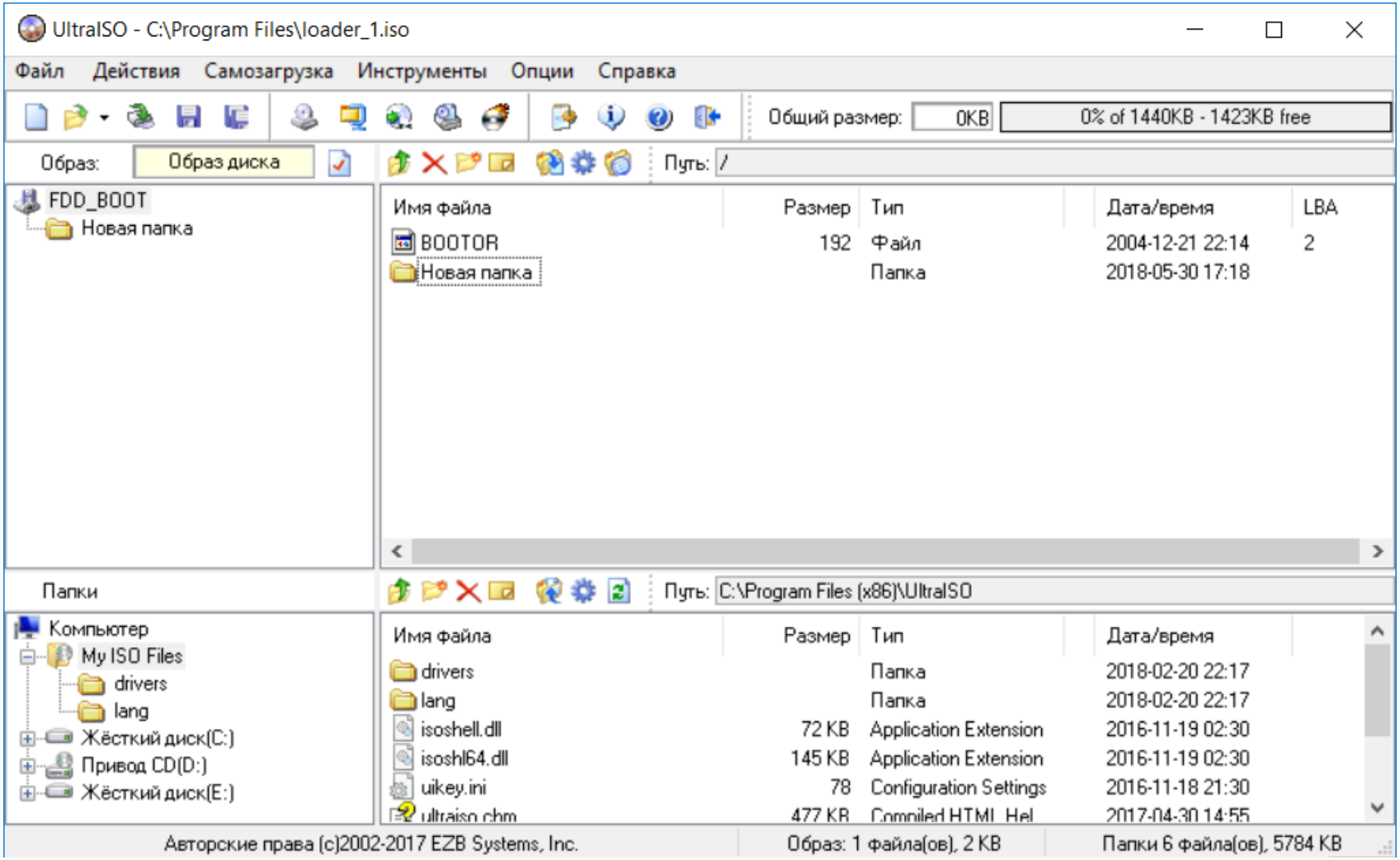
Работа UltraISO



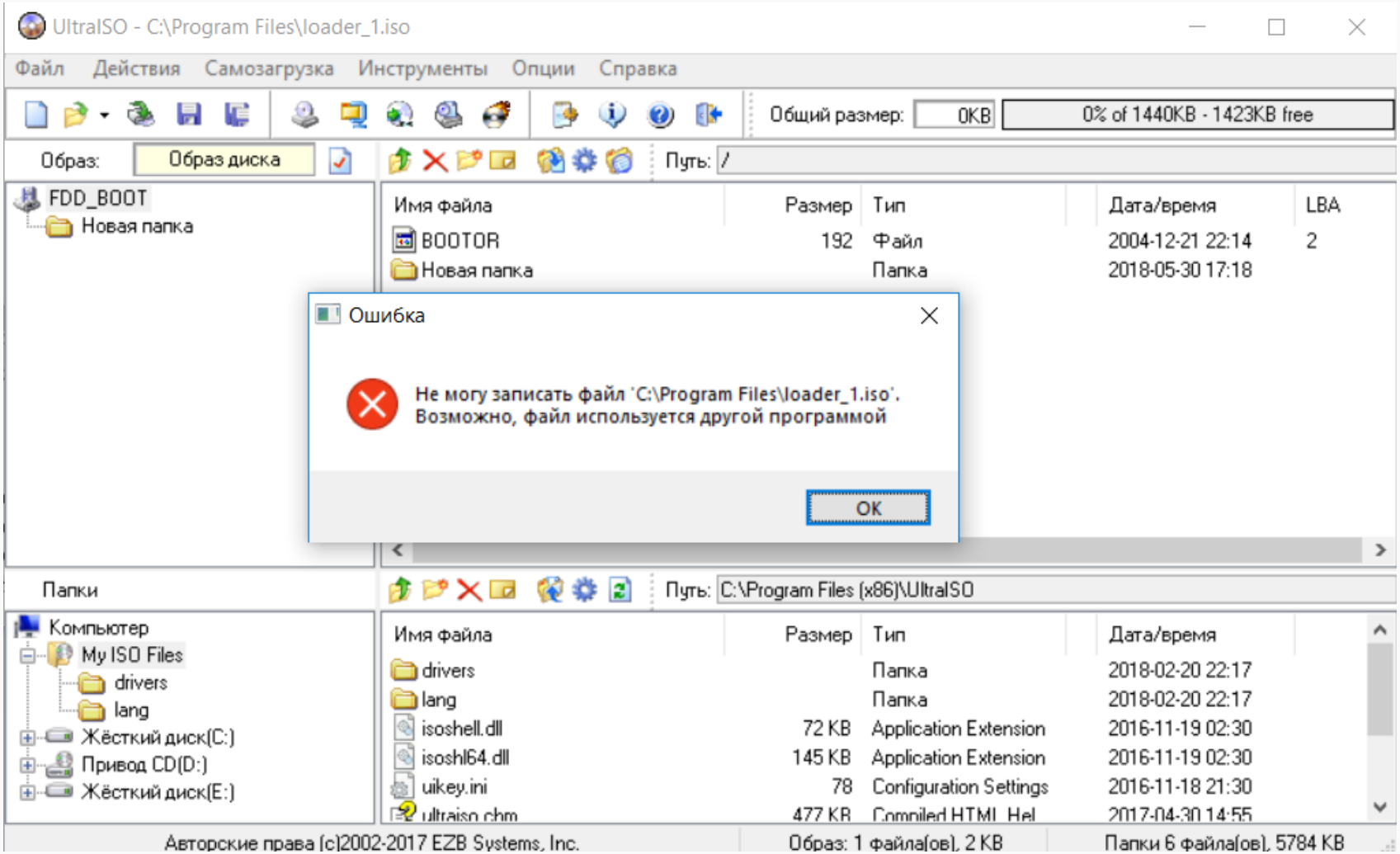
Работа UltraISO



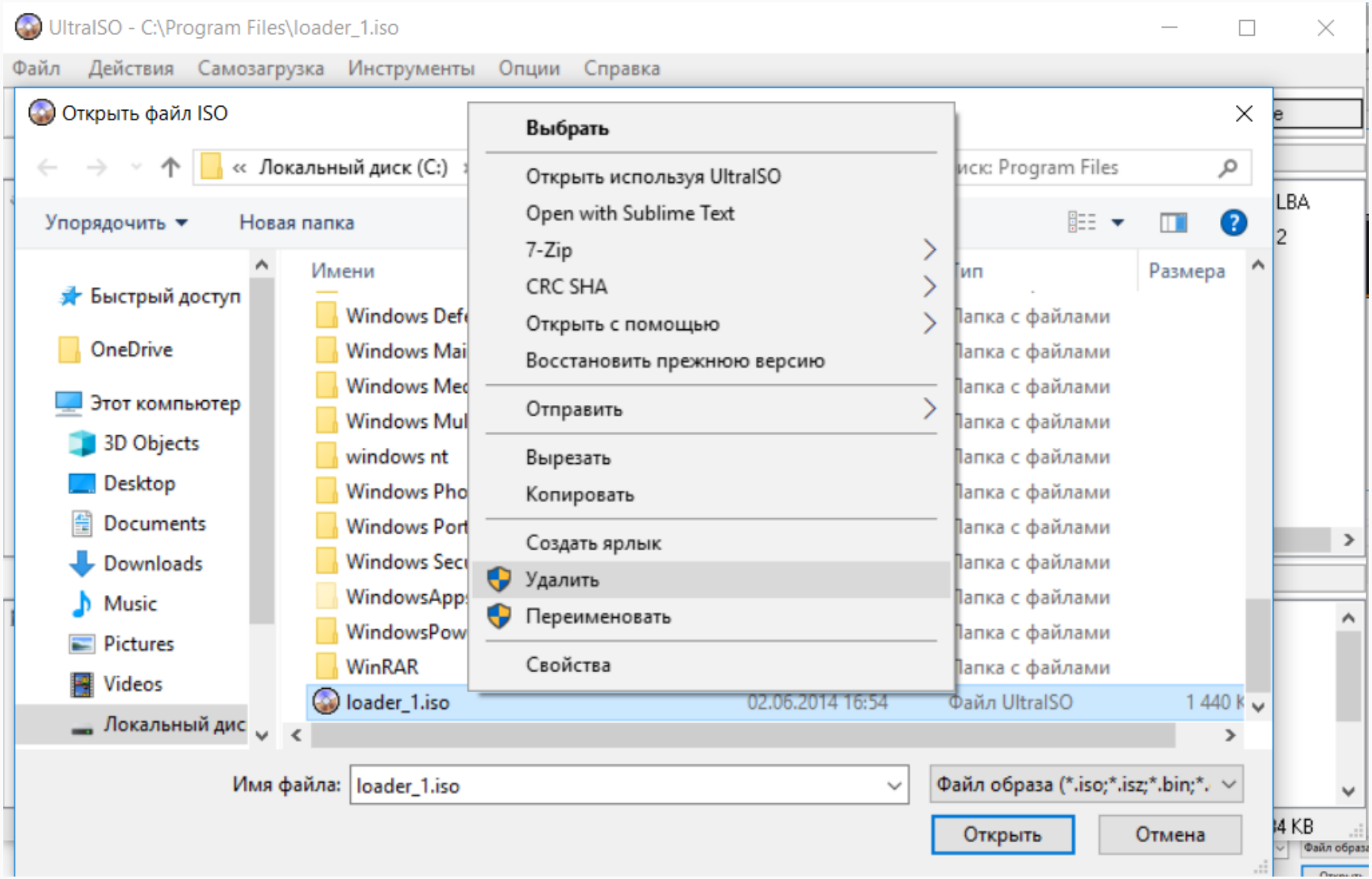
Работа UltraISO



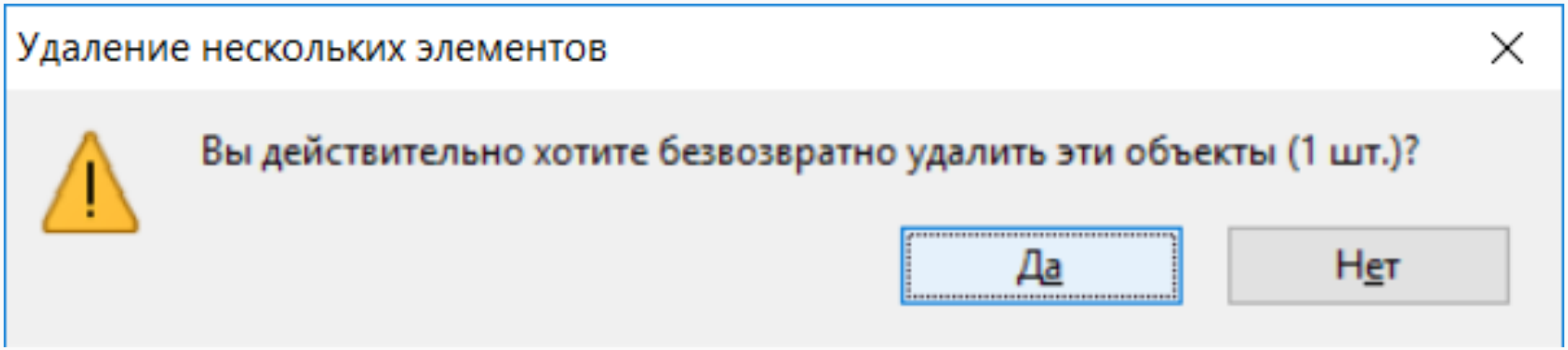
Работа UltraISO



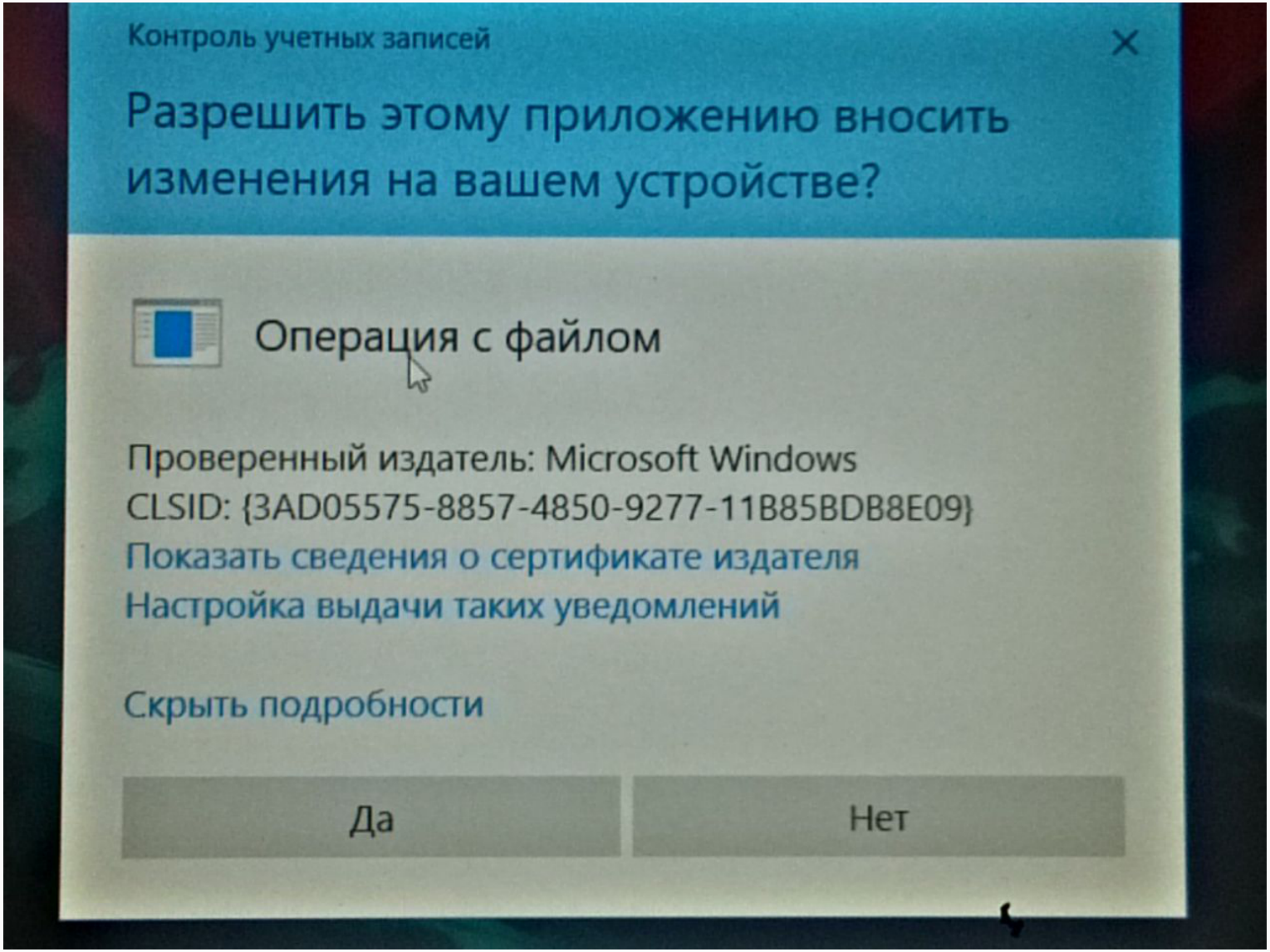
Работа UltraISO



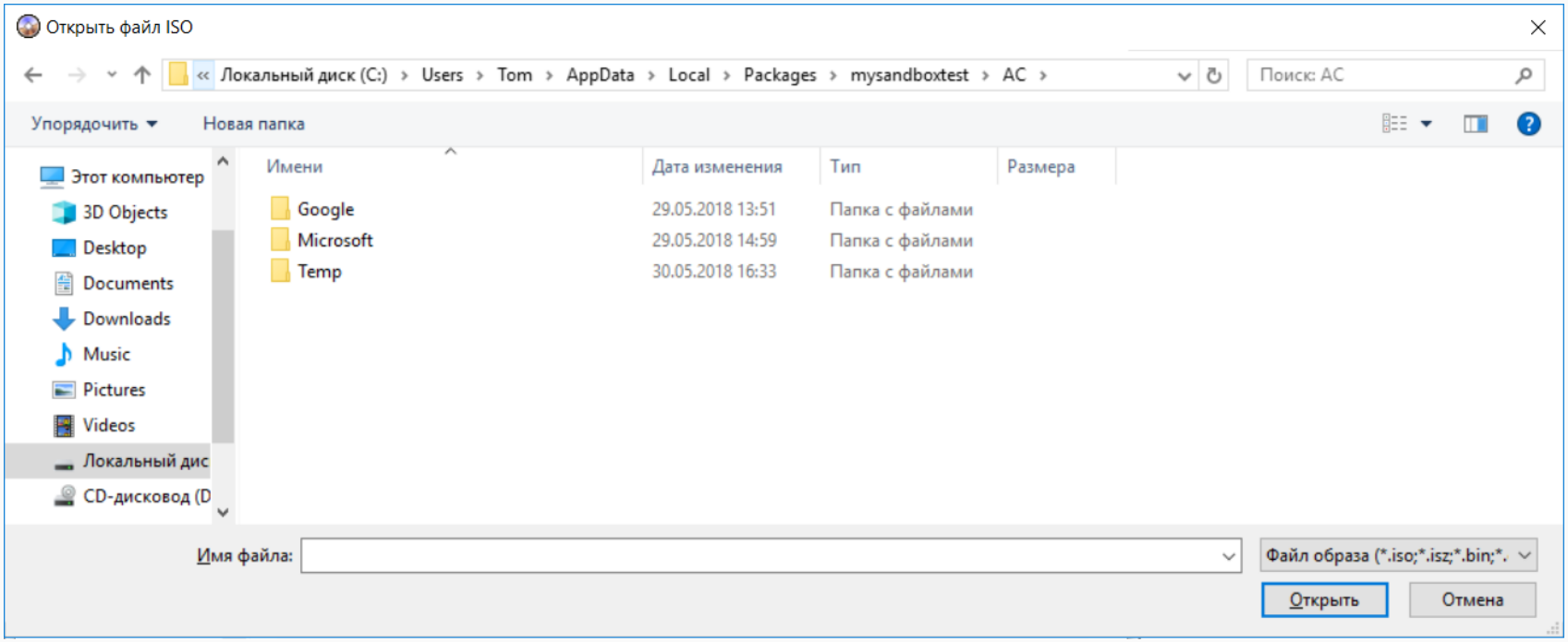
Работа UltraISO



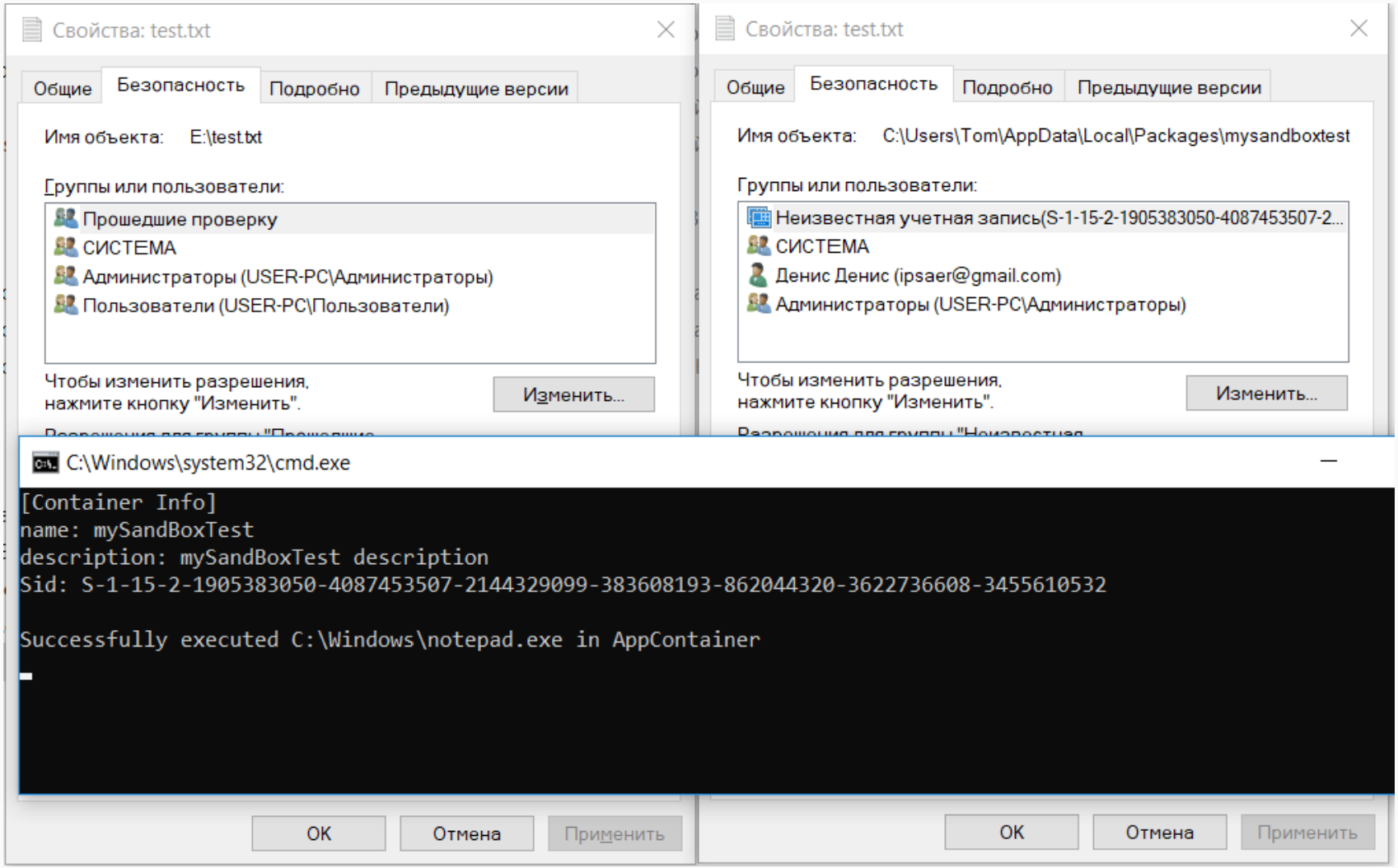
Работа UltraISO



Работа UltraISO



Сравнение свойств файлов



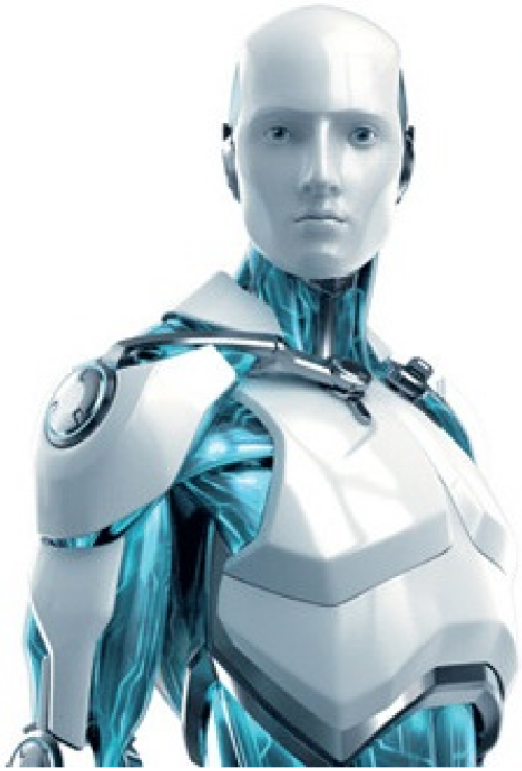
Трудности применения LPAC

- проблемы с GUI(работает несколько иначе);
- запуск любого не примитивного приложения является отдельной, достаточно трудоемкой задачей
  - зависимость от прочих сервисов;
  - зависимость от настроек реестра и прочих файлов (то что происходит во время установки)
- трудность выявления ошибок, многие приложения вовсе не сигнализируют об ошибках;
- LPAC является сравнительно новым инструментом, и слабо документирован.

Потребитель

KASPERSKY lab

avast! AntiVirus



eset

NOD32 ANTIVIRUS

Список использованных источников

- https://msdn.microsoft.com/en-us/library/windows/desktop/hh448541(v=vs.85).aspx
- https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa446585(v=vs.85).aspx
- https://msdn.microsoft.com/en-us/library/windows/desktop/ms686880(v=vs.85).aspx
- https://msdn.microsoft.com/ru-ru/library/windows/desktop/ms724265(v=vs.85).aspx
- https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa446645(v=vs.85).aspx
- https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa379576(v=vs.85).aspx
- https://msdn.microsoft.com/ru-ru/library/windows/desktop/ms686206(v=vs.85).aspx

Список использованных источников

- https://msdn.microsoft.com/ru-ru/library/windows/desktop/ms724265(v=vs.85).aspx
- https://docs.microsoft.com/ru-ru/virtualization/windowscontainers/about/
- https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-190.pdf