1. Расшифруйте аббревиатуру DLL.

Dynamic Link Library

1. Поясните понятия «раннее связывание» и «позднее связывание».

Раннее связывание (early binding):

В случае раннего связывания, связывание данных с кодом происходит на этапе компиляции программы, то есть на этапе, когда программа преобразуется из исходного кода в машинный код.

Позднее связывание (late binding):

В случае позднего связывания, связывание данных с кодом происходит в процессе выполнения программы, а не на этапе компиляции.

1. Как называется функция, которая является точкой входа DLL-библиотеки и в каких случаях эта функция вызывается.

DllMain

1. При первой загрузке DLL, перед вызовом других функций в библиотеке.

2. При каждом выгрузке DLL, перед завершением работы библиотеки.

3. При каждом создании или удалении каждого потока вызывается DllMain с указанием соответствующего события.

Операционная система вызывает функцию точки входа всякий раз, когда происходит одно из следующих событий:

* процесс загружает DLL. Для процессов, использующих динамическое связывание во время запуска программы, DLL загружается во время инициализации процесса. Для процессов, использующих динамическое связывание во время выполнения программы, DLL загружается перед тем, как функция [LoadLibrary](https://docs.microsoft.com/ru-ru/windows/win32/api/libloaderapi/nf-libloaderapi-loadlibraryw" \t "_blank) или [LoadLibraryEx](https://docs.microsoft.com/ru-ru/windows/win32/api/libloaderapi/nf-libloaderapi-loadlibraryexw) возвратит значение [завершит свою работу];
* процесс выгружает DLL. DLL выгружается, когда процесс завершается или вызывает функцию [FreeLibrary](https://docs.microsoft.com/ru-ru/windows/win32/api/libloaderapi/nf-libloaderapi-freelibrary" \t "_blank) и счетчик ссылок по данной DLL становится равным нулю. Если процесс завершается из-за вызова функций [TerminateProcess](https://docs.microsoft.com/ru-ru/windows/win32/api/processthreadsapi/nf-processthreadsapi-terminateprocess" \t "_blank) или [TerminateThread](https://docs.microsoft.com/ru-ru/windows/win32/api/processthreadsapi/nf-processthreadsapi-terminatethread" \t "_blank), операционная система не вызывает функцию точки входа DLL;
* в процессе, который загрузил DLL, создается новый поток выполнения. Можно использовать функцию [DisableThreadLibraryCalls](https://docs.microsoft.com/ru-ru/windows/win32/api/libloaderapi/nf-libloaderapi-disablethreadlibrarycalls" \t "_blank), чтобы отключить уведомления о создании новых потоков выполнения;
* поток выполнения процесса, загрузившего DLL, завершается нормально (то есть не из-за вызова функций [TerminateThread](https://docs.microsoft.com/ru-ru/windows/win32/api/processthreadsapi/nf-processthreadsapi-terminatethread" \t "_blank) или [TerminateProcess](https://docs.microsoft.com/ru-ru/windows/win32/api/processthreadsapi/nf-processthreadsapi-terminateprocess" \t "_blank)). Когда процесс выгружает DLL, функция точки входа вызывается только один раз для всего процесса, а не по разу на каждый существующий поток выполнения процесса. Можно использовать функцию [DisableThreadLibraryCalls](https://docs.microsoft.com/ru-ru/windows/win32/api/libloaderapi/nf-libloaderapi-disablethreadlibrarycalls" \t "_blank), чтобы отключить уведомления о завершении потоков выполнения.

Только один поток выполнения за раз может вызвать функцию точки входа.

1. Какая программа создает DLL?

**Microsoft Visual C++** (MSVC)

1. Для чего применяется директива *EXTERN “C”.*

extern "C" указывает, что функция определена в другом месте и использует соглашение о вызовах на языке C.

1. Для чего экспортируются функции DLL?

Экспорт функций из DLL (Dynamic Link Library) является способом предоставления доступа к функциональности DLL извне, например, из других программ или библиотек. Когда функция экспортируется, она становится доступной для использования другими программами, в том числе исходным кодом, который компилируется отдельно и затем связывается с DLL.

1. Предоставление общего функционала: Многие DLL предоставляют набор общих функций, которые могут быть использованы различными программами. Например, DLL может содержать функции для работы с базами данных, математическими расчетами и т.д.

2. Расширение функционала программы: DLL могут быть использованы для добавления дополнительных возможностей или функций к программе. Например, программа может использовать DLL для добавления поддержки определенных типов файлов, расширенной графики или ввода-вывода.

1. Назовите 2 способа экспорта функций DLL.

1) Использование ключевого слова \_\_declspec(dllexport) перед объявлением каждой экспортируемой функции в исходном коде. Например:

```cpp

\_\_declspec(dllexport) void MyFunction1();

\_\_declspec(dllexport) void MyFunction2();

```

2) Создание файла .def (definition) для указания функций, которые нужно экспортировать. В файле .def перечисляются имена экспортируемых функций. Например:

```

EXPORTS

MyFunction1

MyFunction2

```

1. Как называется функция загрузки DLL? Какие параметры принимает эта функция?

Функция загрузки DLL называется LoadLibrary. Она принимает один параметр - строку, содержащую имя DLL, которую нужно загрузить.

1. Как называется функция выгрузки DLL? Какие параметры принимает эта функция?

Функция выгрузки DLL называется FreeLibrary. Она принимает один параметр - дескриптор загруженной библиотеки (полученный функцией LoadLibrary или LoadLibraryEx).

1. Для чего применяется функция GetProcAddress? Какие параметры принимает эта функция? Что возвращает эта функция?

Функция GetProcAddress используется в операционной системе Windows для получения адреса функции, экспортированной из разделяемой библиотеки (DLL). Это позволяет программе получить доступ к функциям, которые не были изначально загружены при запуске программы, но могут быть загружены динамически по мере необходимости.

Параметры функции GetProcAddress:

1. hModule - дескриптор модуля. Это указатель на загруженную библиотеку (DLL) или исполняемый файл (.exe). Если параметр равен NULL, то функция использует предопределенный контекст, который включает все модули, загруженные в адресное пространство вызывающего процесса, включая исполняемый файл и все DLL.

2. lpProcName - Имя функции, адрес которой нужно получить. Это строка, содержащая имя экспортируемой функции. Функция GetProcAddress зависит от регистра букв в имени.

Возвращаемое значение функции GetProcAddress - адрес функции. Этот адрес можно использовать для вызова функции непосредственно.

1. Поясните выражение «DLL проецируется в адресное пространство процесса».

Выражение «DLL проецируется в адресное пространство процесса» означает, что в процессе загрузки и выполнения программы, динамическая библиотека (DLL - Dynamic Link Library) загружается в адресное пространство процесса, что позволяет программе обращаться к функциям и ресурсам, предоставляемым этой DLL.

Когда процесс загружается в оперативную память компьютера, ему выделяется свое адресное пространство. Это адресное пространство разделено на различные сегменты, такие как код, данные и стек. Когда DLL проецируется в адресное пространство процесса, ее код и данные становятся доступными для использования программой, которая загрузила эту DLL.

Когда программа обращается к функциям из DLL, он использует таблицу адресов функций, известную как таблица импорта. По мере необходимости, эти функции загружаются и преобразуются внутри адресного пространства процесса, и программа может вызывать их, чтобы выполнить определенные задачи.

В конечном итоге, проецирование DLL в адресное пространство процесса позволяет программам использовать общие функции и ресурсы, предоставляемые этой DLL, без необходимости иметь их собственные копии. Это упрощает разработку программ, облегчает обновление и поддержку, а также позволяет более эффективное использование системных ресурсов.

1. Поясните понятие «библиотека импорта».

Библиотека импорта - это набор предопределенных функций и классов, которые могут быть использованы в программировании для выполнения определенных задач без необходимости написания кода с нуля. Библиотеки импорта позволяют программистам использовать уже существующий код, чтобы упростить разработку программы и повысить ее эффективность.

1. Какая программа создает библиотеку импорта.

Программа, которая создает библиотеку импорта, является компилятором языка программирования. Компилятор преобразует исходный код программы на высокоуровневом языке программирования в машинный код, который может быть выполнен процессором компьютера. В процессе компиляции, компилятор также генерирует библиотеку импорта, которая содержит информацию о внешних функциях и методах, которые используются в программе и которые будут вызваны из других модулей или библиотек при выполнении программы.

1. Расшифруйте аббревиатуру COM. Поясните смысл термина.

Common Object Model. COM – модель программного обеспечения. Модель разработана Microsoft. Может применяться во всех операционных системах, но прижилась только в Windows.

1. Поясните понятие «клиент-серверная архитектура приложения».

Будем говорить, что распределенное приложение имеет архитектуру клиент-сервер, если все процессы распределенного приложения можно условно разбить на две группы. Одна группа процессов называется серверами другая – клиентами.

Обмен данными осуществляется только между процессами-клиентами и процессами-серверами.

Основное отличие процесса-клиента от процесса-сервера в том, что инициатором обмена данными всегда является процесс-клиент. Другими словами процесс-клиент обращается за услугой (сервисом) к процессу-серверу

1. Для чего применяется OLE32.DLL?

**COM-клиент - COM-сервер: при создании объекта** посредником между COM-клиентом и COM-сервером выступает библиотека **OLE32.DLL** (библиотека импорта **OLE32.LIB**).



1. Назовите 2 типа контейнера для COM-сервера.

exe и dll

1. Перечислите три типа COM-серверов.

**CLSCTX\_INPROC\_SERVER** (DLL внутрипроцессный сервер); **CLSCTX\_LOCAL\_SERVER** (EXE-сервер за границами процесса, но та том же компьютере); **СLSCTX\_REMOTE\_SERVER** (EXE-сервер на удаленном компьютере).

1. Чем отличаются однокомпонентные сервера от многокомпонентных.

может быть **однокомпонентным** (реализующим один тип объектов) или **многокомпонентным** (реализующим несколько типов объектов).

1. Расшифруйте аббревиатуру GUID, поясните смысл термина.

**GUID(Globally Unique Identifier, 128bit)** – реализация Microsoft стандарта OSF (часть стандарта ISO/IEC 11540 (OSI), ISO/IEC 9834-8 (Object Identifier), RFC 4122 (UUID)) идентификации **UUID** (Universally Unique Identifier).

GUID - тип данных размером 128 бит, который используется для идентификации com-компонета или com-интерфейса

1. Для чего используется макрос STDMETHODCALLTYPE?

STDMETHODCALLTYPE управляет соглашением о вызовах для COM (объектная модель компонентов). Расширяется в \_\_stdcall для x86.

1. Для чего используется структура HRESULT?

HRESULT тип данных который хранит информацию о результате вызова функции компонента

размер 32 бит

первый бит указывает на успешность выполнения функции

следующие 15 бит хранят информацию о типе ошибке

и последние 16 бит хранят специфическую информацию об ошибке

**Все функции OLE32.DLL** возвращают значение типа **HRESULT**.

1. Поясните термин «стандартный CОМ-интерфейс».

"стандартный" com-интерфейс это интерфейс который описан в спецификации COM - все знают его IID

может быть реализован разными компонентами программного обеспечения для обеспечения стандартизированной функциональности

1. Перечислите стандартные COM-интерфейсы и их методы.

IClassFactory:

\* CreateInstance (возвращает экземляр пользовательского компонента)

\* LockServer (запрещает разрушение экземпляра фабрики классов)

IUnknown:

- QueryInterface: используется для получения доступа к другим интерфейсам объекта

- AddRef: увеличивает счетчик ссылок объекта

- Release: уменьшает счетчик ссылок объекта

QueryInterface (находит интерфейсы по IID)

\* AddRef (инкрементирует счетчик ссылок на интерфейс)

\* Release (декрементирует счетчик ссылок на интерфейс)

1. Для чего применяется утилита Regsvr32?

Утилита Regsvr32 используется для регистрации и отмены регистрации динамических ссылок на библиотеки (DLL) в операционной системе Windows. Она позволяет программам получать доступ к функциям и объектам, которые содержатся в этих DLL.

1. Перечислите экспортируемы функции COM-сервера типа INPROC.

1. DllGetClassObject - возвращает указатель на фабрику классов, реализующую интерфейс IClassFactory.

2. DllCanUnloadNow - определяет, можно ли выгрузить сервер из памяти. Возвращает S\_OK, если сервер не используется ни одним клиентом, и S\_FALSE в противном случае.

3. DllRegisterServer - регистрирует COM-сервер в реестре операционной системы.

4. DllUnregisterServer - удаляет регистрацию COM-сервера из реестра операционной системы.

1. Перечислите счетчики, которые должен поддерживать COM-сервер.

???????????????хз

COM-сервер должен поддерживать следующие счетчики:

1. Счетчик ссылок (reference count): этот счетчик отслеживает количество ссылок на объект COM-сервера. Каждый раз, когда на объект создается новая ссылка, счетчик увеличивается, и когда ссылка уничтожается объектом или клиентским приложением, счетчик уменьшается. Когда счетчик достигает нуля, объект освобождается.

2. Счетчик блокировки (lock count): этот счетчик используется для блокировки объекта COM-сервера, чтобы предотвратить доступ других клиентских приложений к объекту во время выполнения операции. Когда объект блокируется, счетчик увеличивается, и когда разблокируется, счетчик уменьшается. Обычно объект блокируется только для операций, которые изменяют его состояние.

3. Счетчик ошибок (error count): этот счетчик отслеживает количество ошибок, произошедших в объекте COM-сервере. Когда происходит ошибка, счетчик увеличивается, и клиентское приложение может получить информацию о произошедшей ошибке.

4. Счетчик вызовов (call count): этот счетчик отслеживает количество вызовов методов объекта COM-сервера. Каждый раз, когда клиентское приложение вызывает метод объекта, счетчик увеличивается.

Это основные счетчики, которые требуется поддерживать в COM-сервере, но в зависимости от конкретной реализации и требований могут быть и другие счетчики.

1. Назначение метода QueryInterface.

используется для получения доступа к другим интерфейсам объекта

находит интерфейсы по IID

1. Назначение метода AddRef.

инкрементирует счетчик ссылок на интерфейс

1. Назначение метода Release.

декрементирует счетчик ссылок на интерфейс

1. Назначение метода CreateInstance.

CreateInstance (возвращает экземляр пользовательского компонента)

1. Назначение метода ServerLock.

LockServer (запрещает разрушение экземпляра фабрики классов)

1. Напишите три условия при котором DllCanUnloadNow возвращает Succeeded-значение.

DllCanUnloadNow возвращает значение Succeeded (S\_OK) в следующих условиях:

1. Если в DLL нет активных экземпляров или ссылок на объекты. Это означает, что все созданные объекты уже были освобождены и нет открытых ссылок на них.

2. Если в DLL нет активных потоков выполнения. DLL может быть выгружена только в том случае, если все потоки, которые могут использовать ресурсы DLL, завершены.

3. Если есть соответствующее управление жизненным циклом DLL. DLL не должна содержать зависимостей от ресурсов, которые могут оставаться активными во внешнем контексте (например, открытые файлы, сокеты и т. д.).

1. Что функция DllGetClassObject возвращает последним параметром?

???????????????

Функция DllGetClassObject возвращает последним параметром указатель на перменную типа "IClassFactory".

1. Каким способом вызываются функции DllRegisterServer, DllUnregisterServer и DllInstall?

DllRegisterServer:

Эта функция вызывается при регистрации DLL с помощью утилиты regsvr32 или аналогичных инструментов. Обычно DllRegisterServer содержит код для добавления записей в реестр, указывающих на расположение и атрибуты COM-объектов, предоставляемых DLL.

DllUnregisterServer:

Эта функция вызывается при отмене регистрации DLL, например, с использованием regsvr32 /u. DllUnregisterServer обычно содержит код для удаления записей из реестра, связанных с COM-объектами DLL.

DllInstall:

Эта функция вызывается для установки или отмены установки COM-объектов из DLL. Она может быть использована для проведения различных настроек или операций в зависимости от требований приложения. Параметры функции могут включать информацию о регистрации или отмене регистрации, а также другие данные, необходимые для настройки COM-объектов.