**Коллоквиум 1, СП-ПОИТ-3-2023**

***Вопросы не пишем, пишем номер и ответ***.

1. **В правом верхнем углу листа:**

***1-СП-ПОИТ-3/группа, Фамилия И.О.***

1-СП-ПОИТ-3/4, Поздняков М. И.

1. **Расшифруйте аббревиатуру DLL.**

Dynamic Link Library – библиотека динамической компоновки.

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

Раннее связывание означает, что объект и вызов функции связываются между собой на этапе компиляции. Это означает, что вся необходимая информация для того, чтобы определить, какая именно функция будет вызвана, известна на этапе компиляции программы. Адрес вызываемой функции определяется на этапе компиляции и остается неизменным во время выполнения программы.

Достоинство: эффективность — оно бо­лее быстрое и обычно требует меньше памяти, чем позднее связывание.

Недостаток: невысокая гибкость.

Позднее связывание означает, что объект связывается с вызовом функции только во время исполнения программы, а не раньше. Адрес вызываемой функции определяется во время выполнения программы, что позволяет использовать полиморфизм и обеспечивает большую гибкость.

Достоинство: высокая гибкость

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

DllMain

Вызывается в следующих случаях:

1) При подключении библиотеки:

Когда процесс загружает DLL с помощью функции LoadLibrary.

2) При отключении библиотеки:

Когда процесс выгружает DLL с помощью функции FreeLibrary.

3) При запуске нового потока в процессе:

Когда создается поток в процессе, в который была загружена DLL.

4) При завершении существующего потока в процессе:

Когда завершается поток в процессе, в который была загружена DLL.

1. **Какая программа создает DLL?**

DLL создается с использованием компилятора, который поддерживает создание динамических библиотек.( Microsoft Visual Studio, MinGW).

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

Директива extern "C" применяется в языке программирования C++ и используется для указания компилятору, что следующий блок кода должен быть компилирован с использованием соглашений о вызовах, принятых в языке C.

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

Когда функции или объекты классов в C++ объявляются с использованием extern "C", компилятор C++ отключает C++-специфичные функции, такие как name mangling (декорирование имен), что делает интерфейс более совместимым с кодом, написанным на языке C.

Без extern "C" компилятор C++ может изменить имена функций (мэнглировать их), что может привести к проблемам при попытке вызова этих функций из кода на C.

1. **Для чего экспортируются функции DLL?**

Экспорт функций из DLL (Dynamic Link Library) позволяет другим программам или модулям использовать эти функции. Экспортированные функции предоставляют интерфейс для взаимодействия с кодом, находящимся внутри DLL.

1. **Назовите 2 способа экспорта функций DLL.**

1) Использование ключевых слов и атрибутов компилятора:

Использование ключевого слова \_\_declspec(dllexport):

\_\_declspec(dllexport) int MyExportedFunction(int param);

2) Использование DEF-файла (Module-Definition File):

Файл с расширением .def (DEF-файл), в котором перечислены все функции подлежащие экспорту.

LIBRARY MyLibrary

EXPORTS

MyExportedFunction

Затем DEF-файл используется при компоновке DLL для указания компилятору, какие функции экспортировать.

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

HMODULE LoadLibraryA(LPCSTR lpLibFileName)

ИЛИ

HMODULE LoadLibraryW(LPCWSTR lpLibFileName)

Она находится в библиотеке kernel32.dll и предоставляет возможность загрузить динамическую библиотеку в память и получить дескриптор модуля.

lpLibFileName: Указатель на строку с именем модуля, который надо загрузить. Это может быть как библиотека (.dll), так и исполняемый модуль (.exe). Это может быть абсолютный или относительный путь к файлу. Функция и параметр отличается в зависимости от кодировки строки ANSI или UNICODE.

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

BOOL FreeLibrary(HMODULE hLibModule);

hLibModule: Дескриптор загруженного модуля (HMODULE), который был получен при вызове функции LoadLibrary.

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

Извлекает адрес экспортированной функции или переменной из указанной библиотеки динамической компоновки (DLL).

FARPROC GetProcAddress([in] HMODULE hModule, [in] LPCSTR lpProcName);

hModule: Дескриптор модуля (HMODULE), представляющий загруженную DLL, из которой нужно получить адрес функции или переменной.

lpProcName: Строка, содержащая имя функции или переменной, адрес которой вы хотите получить.

Если функция завершается успешно, возвращаемым значением является адрес экспортируемой функции или переменной. Если функция завершается неудачно, возвращаемое значение равно NULL.

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

Выражение "DLL проецируется в адресное пространство процесса" означает, что динамическая библиотека (DLL) загружается в оперативную память процесса и становится доступной для исполнения программы, работающей в этом процессе.

Проецирование DLL в адресное пространство процесса обеспечивает возможность динамической загрузки кода и данных во время выполнения программы, что обеспечивает гибкость и переиспользование кода.

1. **Поясните понятие «библиотека импорта».**

Библиотека импорта для dll - это вспомогательный артефакт связывания основного образа программы с динамическим модулем.

Библиотека импорта (Import Library) обычно имеет расширение .lib. Это стандартное расширение для статических библиотек в системах Windows. Файл библиотеки импорта содержит информацию о функциях или переменных, которые используются в программе, но реализация которых предоставляется внешней DLL (динамической библиотеке).

1. **Какая программа создает библиотеку импорта.**

Библиотеку импорта создает компилятор в процессе компиляции программы, которая использует функции из внешних DLL. Программа компилирует исходный код и создает объектные файлы, включая библиотеку импорта.

1. **Расшифруйте аббревиатуру COM. Поясните смысл термина.**

COM – Component Object Model (модель компонентного объекта), модель проектирования и создания компонентных модулей, в которой соблюдается определенный двоичный стандарт. Этот двоичный стандарт обеспечивает средства, с помощью которых объекты и компоненты, разработанные на разных языках программирования разными поставщиками и работающие в различных операционных системах, могут взаимодействовать без каких-либо изменений в двоичном (исполняемом) коде.

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

Приложение (программа) с клиент-серверной архитектурой состоит из двух компонент – клиента и сервера; клиент и сервер взаимодействуют между собой в соответствии с заданными правилами (протоколами); должно быть установлено соединение.

1. **Для чего применяется OLE32.DLL?**

Обеспечивает работу COM-приложений.

OLE32.DLL (Object Linking and Embedding 32-bit Dynamic Link Library) — это динамическая библиотека, используемая в операционных системах Microsoft Windows для реализации технологии Object Linking and Embedding.

1. **Назовите 2 типа контейнера для COM-сервера.**

Для размещения компонентов могут быть применены два вида контейнеров: DLL-файл и EXE-файл.

1. **Перечислите три типа COM-серверов.**

CLSCTX\_INPROC\_SERVER - dll-сервер внутрипроцессовый

CLSCTX\_LOCAL\_SERVER - exe-сервер, который работает за пределами процесса, но на той же машине

СLSCTX\_REMOTE\_SERVER - exe-сервер, который работает на удаленной машине

1. **Чем отличаются однокомпонентные сервера от многокомпонентных.**

Однокомпонентный – реализующим один тип объектов.

Многокомпонентным – реализующим несколько типов объектов.

1. **Расшифруйте аббревиатуру GUID, поясните смысл термина.**

Используемый в СОМ GUID (Globally Unique Identifier – глобальный уникальный идентификатор) представляет собой 128-разрядное значение, которому гарантируется статистическая уникальность. В СОМ идентификаторы GUID используются для идентификации классов компонента (CLSID), интерфейсов (IID).

1. **Для чего используется макрос STDMETHODCALLTYPE?**

Расширяется в \_\_stdcall, указывает что будет использоваться соглашение о вызове stdcall (аргументы передаются через стек, справа налево, очистку стека производит вызываемая подпрограмма).

1. **Для чего используется структура HRESULT?**

32-битное значение, которое используется для описания ошибок. В COM используется как возвращаемое значение всех методов (как стандартных, так и реализованных пользователем), кроме AddRef() и Release().

1. **Поясните термин «стандартный CОМ-интерфейс».**

За стандартными интерфейсами закреплены предопределенные GUID идентификаторы. Т.е. стандартный COM-интерфейс это тот, чей IID знают все.

1. **Перечислите стандартные COM-интерфейсы и их методы.**

IUnknown -> QueryInterface(), AddRef(), Release().

IClassFactory -> CreateInstance(), LockServer().

1. **Для чего применяется утилита Regsvr32?**

Regsvr32 — это служебная программа командной строки для регистрации и отмены регистрации элементов управления OLE, например библиотеки DLL в реестре Windows.

1. **Перечислите экспортируемы функции COM-сервера типа INPROC.**

DllRegisterServer, DllUnregisterServer, DllInstall, DllGetClassObject, DllCanUnloadNow.

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

Счетчик блокировок, счетчик ссылок на интерфейсы, счетчик ссылок на экземпляр компонентов.

1. **Назначение метода QueryInterface.**

HRESULT QueryInterface (REFIID riid, void \*\*ppvObject);

Запрашивает у COM-объекта указатель и записывает адрес в ppvObject на один из его интерфейсов, IID, которому соответствует riid.

1. **Назначение метода AddRef.**

ULONG AddRef();

Инкрементирует счетчик ссылок на интерфейс COM-объекта и возвращает текущее значение.

1. **Назначение метода Release.**

ULONG Release();

Декрементирует счетчик ссылок на интерфейс COM-объекта и освобождает объект. Возвращает текущее значение.

1. **Назначение метода CreateInstance.**

HRESULT CreateInstance([in] IUnknown \*pUnkOuter, [in] REFIID riid, [out] void \*\*ppvObject);

Создает и возвращает ссылку на экземпляр пользовательского компонента.

1. **Назначение метода ServerLock.**

HRESULT LockServer([in] BOOL fLock);

Запрещает разрушение экземпляра фабрики классов.

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

Счетчик объектов ноль, счетчик блокировок ноль, не используется в агрегации.

1. **Что функция DllGetClassObject возвращает последним параметром?**

HRESULT DllGetClassObject([in] REFCLSID rclsid, [in] REFIID riid, [out] LPVOID \*ppv);

ppv - Адрес переменной-указателя, которая получает указатель интерфейса, запрошенный в riid. При успешном возврате \* ppv содержит запрошенный указатель интерфейса. В случае возникновения ошибки указатель интерфейса равен NULL.

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

Используются regsvr32.

DllInstall и DllRegisterServer – ключом /i (/n – не вызывает DllRegisterServer, может использоваться с ключом /i).

DllUnregisterServer – ключом /u.