32

Межсистемные интерфейсы и драйверы: интерфейсы в распределенных системах.

**Распределённая система** — система, для которой отношения местоположений элементов (или групп элементов) играют существенную роль с точки зрения функционирования системы, а следовательно, и с точки зрения анализа и синтеза системы.

Для распределённых систем характерно распределение функций, ресурсов между множеством элементов (узлов) и отсутствие единого управляющего центра, поэтому выход из строя одного из узлов не приводит к полной остановке всей системы. Типичной распределённой системой является Интернет.

**В чем разница между централизованной и распределенной системами?**

В централизованной вычислительной системе все вычисления выполняются на одном компьютере и в одном месте. Основное различие между централизованными и распределенными системами заключается в модели взаимодействия между узлами системы. Состояние централизованной системы хранится в центральном узле, к которому индивидуально обращаются клиенты. Поскольку все узлы централизованной системы обращаются к центральному узлу, это может привести к перегрузке сети и замедлить ее работу. Централизованная система имеет единую точку отказа, тогда как в распределенной системе такой точки нет.

**Отличаются ли распределенные системы от микросервисов?**

Микросервисная архитектура — это лишь один из типов распределенных систем. В ней приложение разбивается на отдельные компоненты, или «сервисы». К примеру, микросервисная архитектура может иметь сервисы, соответствующие бизнес-функциям (платежи, пользователи, продукты и др.), и каждый компонент будет обрабатывать бизнес-логику в своей сфере ответственности. В этом случае в системе будет несколько резервных копий сервисов, и у сервисов не будет единой точки отказа.

**Примеры распределённых систем**

1. Распределенная система компьютеров — компьютерная сеть.
2. Распределённая система управления — система управления технологическим процессом.
3. Распределённая энергетика
4. Распределённая экономика
5. Распределённая файловая система — сетевые файловые системы.
6. Распределённые операционные системы
7. Системы распределённых вычислений
8. Распределённые системы контроля версий
9. Распределённые базы данных
10. Система доменных имён (DNS) — распределённая система для получения информации о доменах.

**Интерфейс**

В наиболее общем смысле интерфейсом называется общая граница, через которую передаётся информация (стандарт ISO 24765).

В вычислительной системе взаимодействие может осуществляться на пользовательском, программном и аппаратном уровнях.

Способ взаимодействия физических устройств

Переписать

Этот раздел должен быть полностью переписан.

На странице обсуждения могут быть пояснения.

См. также: Протокол передачи данных и Список пропускных способностей интерфейсов передачи данных

Физический (аппаратный интерфейс) — способ взаимодействия физических устройств. Чаще всего речь идёт о компьютерных портах (разъёмах).

Сетевой интерфейс

Сетевой шлюз — устройство, соединяющее локальную сеть с более крупной, например, Интернетом

Шина (компьютер)

Стандартный интерфейс — совокупность унифицированных технических, программных и конструктивных средств, основанных на стандарте, реализующих взаимодействие различных функциональных элементов в информационной системе, обеспечивающих информационную, электрическую и конструктивную совместимость этих элементов. Стык — место соединения устройств сети передачи данных. Связь между понятиями протокол и интерфейс не всегда однозначна: интерфейс может содержать элементы протокола, а протокол, в свою очередь, может охватывать несколько интерфейсов (стыков). Основная идея использования стандартных интерфейсов и протоколов — унификация меж- и внутрисистемных и меж- и внутрисетевых связей для повышения эффективности проектирования вычислительных систем.

Способ взаимодействия программных компонентов

Основная статья: Программный интерфейс

Прикладной программный интерфейс (API) — набор стандартных библиотечных методов, которые программист может использовать для доступа к функциональности другой программы.

Удалённый вызов процедур

COM-интерфейс

Интерфейс объектно-ориентированного программирования — описание методов взаимодействия объектов приложения на уровне исходного кода

Запись голосовой команды в мобильном приложении или веб-браузере информационных систем; дополнение аудиозаписи идентификатором и прочими метаданными; передача в Интеграционную шину Ассистента речевого управления произвольным интерфейсом; получение от Интеграционной шины идентификатора распознанной команды и её параметра; отправка и исполнение распознанной голосовой команды управления веб-интерфейсом на стороне информационной системы.[6]

Через графический Web-интерфейс, имеющий картографическую основу и позволяющий визуализировать прием, обработку, регистрацию и передачу данных, обеспечивая предоставление цифровых сервисов. Область применения: информационное обеспечение и взаимодействие судов и береговых систем мониторинга и управления. Функциональные возможности: сопряжение с сервисами e-Навигации; обеспечение интерактивной работы с сервисами e-Навигации; отображение данных на электронной навигационной карте; предоставление пользователю необходимых инструментов для работы с сервисами е-Навигации.

Способ взаимодействия человека и техники

Основные статьи: Человеко-машинный интерфейс и Человеко-компьютерное взаимодействие

Человеко-компьютерное взаимодействие (ЧКИ или HCI - human-computer interaction) — полидисциплинарное научное направление, существующее и развивающееся в целях совершенствования методов разработки, оценки и внедрения интерактивных компьютерных систем, предназначенных для использования человеком, а также в целях исследования различных аспектов этого использования.[8]

Интерфейс пользователя: совокупность средств, при помощи которых пользователь взаимодействует с различными программами и устройствами:

Интерфейс командной строки: инструкции компьютеру даются путём ввода с клавиатуры текстовых строк (команд).

Графический интерфейс пользователя: программные функции представляются графическими элементами экрана, WIMP

SILK-интерфейс (от speech — речь, image — образ, language — язык, knowledge — знание): взаимодействие с компьютером посредством речи.

Жестовый интерфейс: сенсорный экран, руль, джойстик и т. д.

Нейрокомпьютерный интерфейс: отвечает за обмен между нейронами и электронным устройством при помощи специальных имплантированных электродов.

**Драйвер** (англ. driver, мн. ч. дра́йверы) — компьютерное программное обеспечение, с помощью которого другое программное обеспечение (операционная система) получает доступ к аппаратному обеспечению некоторого устройства. Обычно с операционными системами поставляются драйверы для ключевых компонентов аппаратного обеспечения, без которых система не сможет работать. Однако для некоторых устройств (таких, как видеокарта или принтер) могут потребоваться специальные драйверы, обычно предоставляемые производителем устройства.

В общем случае драйвер не обязан взаимодействовать с аппаратными устройствами, он может их только имитировать (например, драйвер принтера, который записывает вывод из программ в файл), предоставлять программные сервисы, не связанные с управлением устройствами (например, /dev/zero в Unix, который только выдаёт нулевые байты), либо не делать ничего (например, /dev/null в Unix и NUL в DOS/Windows).

**Подход к построению драйверов**

Операционная система управляет некоторым «виртуальным устройством», которое понимает стандартный набор команд. Драйвер переводит эти команды в команды, которые понимает непосредственно устройство. Эта идеология называется «абстрагирование от аппаратного обеспечения». Впервые в отечественной вычислительной технике подобный подход появился в серии ЕС ЭВМ, а такого рода управляющее программное обеспечение называлось канальным программным обеспечением.

Драйвер состоит из нескольких функций, которые обрабатывают определенные события операционной системы. Обычно основные события следующие:

Загрузка драйвера: драйвер регистрируется в системе, производит первичную инициализацию и т. п.

Выгрузка: драйвер освобождает захваченные ресурсы — память, файлы, устройства и т. п.

Открытие драйвера: обычно драйвер открывается программой как файл, функциями fopen() в UNIX-подобных системах или CreateFile() в Win32. С этого события начинается основная работа драйвера.

Чтение/Запись: программа осуществляет обмен данными с устройством, обслуживаемым драйвером.

Закрытие: операция, обратная открытию, освобождает занятые при открытии ресурсы и уничтожает дескриптор файла.

Управление вводом-выводом (англ. IO Control, IOCTL). Зачастую драйвер поддерживает интерфейс ввода-вывода, специфичный для данного устройства. С помощью этого интерфейса программа может послать специальную команду, которую поддерживает данное устройство. Например, для SCSI-устройств можно послать команду GET\_INQUIRY, чтобы получить описание устройства. В Win32-системах управление осуществляется через API-функцию DeviceIoControl(), в UNIX-подобных — с помощью ioctl().

**Разработка драйвера** для использования не по прямому назначению

Поскольку в ОС Windows драйвер не обязан взаимодействовать с каким-либо внешним устройством, широко применяется разработка "псевдодрайверов", которые значительно расширяют функциональность программы.

Например, большинство современных игровых анти-читов используют драйвер-модуль для более широкого "наблюдения" за всеми процессами, которые происходят в системе. Этот же метод используется разработчиками вредоносного ПО (читов) для обхода защиты.

**Интеграция драйверов**

По мере развития систем, сочетающих в себе на одной плате не только центральные элементы компьютера, но и большинство устройств компьютера в целом, возник вопрос удобства поддержки таких систем, получивших название «аппаратная платформа», или просто «платформа».

Сначала производители платформ поставляли набор отдельных драйверов для операционных систем, собранный на один носитель (обычно компакт-диск), затем появились установочные пакеты, называвшиеся 4-in-1 и One touch, и позволявшие упростить установку драйверов в систему. При этом, как правило, можно выбрать либо полностью автоматическую установку всех драйверов, либо выбрать вручную нужные. Однако единого, устоявшегося термина долго не было.

Современный термин — Board Support Package (или «пакет поддержки платформы»), описывающий такие наборы драйверов устройств. Помимо собственно драйверов, он может, как и прочие установочные пакеты, содержать модули операционной системы и программы.

**Примеры интерфейсов в программировании**

API (application programming interface): интерфейс программирования приложений — набор готовых классов, процедур функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах. API определяет функциональность, которую предоставляет программа (модуль, библиотека), при этом API позволяет абстрагироваться от того, как именно эта функциональность реализована.

Программные компоненты взаимодействуют друг с другом посредством API. При этом обычно компоненты образуют иерархию — высокоуровневые компоненты используют API низкоуровневых компонентов, а те, в свою очередь, используют API ещё более низкоуровневых компонентов.

API библиотеки функций и классов включает в себя описание сигнатур и семантики функций.

COM (Component Object Model — Объектная Модель Компонентов) — это технологический стандарт от компании Microsoft, предназначенный для создания ПО на основе взаимодействующих распределённых компонентов, каждый из которых может использоваться во многих программах одновременно. Стандарт воплощает в себе идеи полиморфизма и инкапсуляции объектно-ориентированного программирования. Стандарт COM закрепился в основном на операционных системах семейства Microsoft Windows. В современных версиях Windows COM используется очень широко. На основе COM были реализованы технологии: Microsoft OLE Automation, ActiveX, DCOM, COM+, DirectX, XPCOM. Стандарт COM был разработан в 1993 году корпорацией Microsoft как основа для развития технологии OLE.

OLE (Object Linking and Embedding) — технология связывания и внедрения объектов в другие документы и объекты, разработанные корпорацией Microsoft. OLE позволяет передавать часть работы от одной программы редактирования к другой и возвращать результаты назад. Программа, которую пытаются внедрить, должна присутствовать на машине клиента. OLE используется при обработке составных документов (compound documents), может быть использована при передаче данных между различными несвязанными между собой системами посредством интерфейса переноса ( drag-and-drop), а также при выполнении операций с буфером обмена.

ActiveX : В 1996 году Microsoft переименовала технологию OLE 2.0 в ActiveX. Были представлены элементы управления ActiveX, ActiveX документы и технология Active Scripting. Эта версия OLE в основном используется веб-дизайнерами для вставки в страницы мультимедийных данных.

TCP: (Transmission Control Protocol) -протокол управления передачей) - один из основных сетевых протоколов Интернета, предназначенный для управления передачей данных в сетях и подсетях TCP/IP. Выполняет функции протокола транспортного уровня модели OSI. TCP — это транспортный механизм, предоставляющий поток данных, с предварительной установкой соединения, за счёт этого дающий уверенность в достоверности получаемых данных, осуществляет повторный запрос данных в случае потери данных и устраняет дублирование при получении двух копий одного пакета. Реализация TCP, как правило, встроена в ядро системы, хотя есть и реализации TCP в контексте приложения. TCP осуществляет надежную передачу потока байтов от одной программы на некотором компьютере к другой программе на другом компьютере. Программы для электронной почты и обмена файлами используют TCP. TCP контролирует длину сообщения, скорость обмена сообщениями, сетевой трафик.

ADO (ActiveX Data Objects —объекты данных ActiveX) — интерфейс программирования приложений для доступа к данным, разработанный компанией Microsoft (MS Access, MS SQL Server) и основанный на технологии компонентов ActiveX. ADO позволяет представлять данные из разнообразных источников (реляционных баз данных, текстовых файлов и т. д.) в объектно-ориентированном виде. Библиотека ADO служит для доступа к базам данных различных типов и предоставляет объектный программный интерфейс. Объектная модель ADO реализована на базе технологии COM (Component Object Model). С помощью библиотеки ADO можно обратиться к огромному количеству типов баз данных, например, dBASE, Access, Excel, Oracle, Paradox, MS SQL Server, Sybase, текстовые файлы, FoxPro, Active Directory Service, Microsoft Jet, Interbase, Informix, PostgreSQL, MySQL и т.д., необходимо только наличие установленного соответствующего OLE-провайдера (драйвера соответствующего типа базы данных, который устанавливается в систему как правило из дистрибутива этой же базы данных).

.NET Framework – это «каркас» программной среды; является открытой инфраструктурой приложения. Может включать вспомогательные программы, библиотеки кода, язык сценариев и другое ПО, облегчающее разработку и объединение разных компонентов большого программного проекта. Обычно объединение происходит за счёт использования единого API. Одной из основных идей Microsoft .NET является совместимость различных служб, написанных на разных языках.

Платформа .NET Framework — это интегрированный компонент Windows, который поддерживает создание и выполнение нового поколения приложений и веб-служб XML.