

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Сибирский государственный университет науки и технологий имени  
академика М.Ф. Решетнева»  
Институт инженерной экономики  
институт  
Кафедра информационно-экономических систем  
кафедра

ДОКЛАД  
Межсистемные интерфейсы и драйверы: интерфейсы в распределённых  
системах  
тема

Преподаватель	<hr/>	К.В. Романов
	подпись, дата	фамилия, инициалы
Обучающийся <u>БПЦ21-01, 211519021</u>	<hr/>	Л.С. Цветков
номер группы, зачётной книжки	подпись, дата	фамилия, инициалы
Обучающийся <u>БПЦ21-01, 211519012</u>	<hr/>	Е.А. Семенов
номер группы, зачётной книжки	подпись, дата	фамилия, инициалы

Красноярск 2024

# Содержание

1	Что такое распределённая система? . . . . .	3
2	В чем разница между централизованной и распределённой системами? . . . . .	3
3	Есть ли отличия у распределённой вычислительной системы от микросервисов? . . . . .	3
4	Примеры распределённых систем. . . . .	4
5	Интерфейс в распределённых системах. . . . .	4
6	Способ взаимодействия физических устройств в распределённых системах. . . . .	4
7	Способ взаимодействия программных компонентов. . . . .	5
8	Способ взаимодействия человека и техники. . . . .	5
9	Драйвер. . . . .	6
10	Подход к построению драйверов. . . . .	6
11	Разработка драйвера. . . . .	7
12	Интеграция драйверов. . . . .	7
13	Примеры интерфейсов в программировании. . . . .	8

## 1 Что такое распределённая система?

Распределённая система — это система, для которой отношения местоположений элементов или групп элементов играют существенную роль с точки зрения функционирования, 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. Например, распределённой системой является Интернет.

## 2 В чем разница между централизованной и распределённой системами?

В централизованной вычислительной системе все вычисления происходят в одном месте на одном устройстве. Главное отличие от распределённых вычислительных систем — модель взаимодействия между узлами. Состояние системы хранится в центральном узле, к которому обращаются все клиенты. Из-за этого сеть может перегружаться, а работа системы — замедляться. Еще один минус — единая точка отказа, отсутствующая в распределённых системах.

## 3 Есть ли отличия у распределённой вычислительной системы от микросервисов?

Микросервисная архитектура — разновидность распределённых вычислительных систем, где приложение разбито на самостоятельные компоненты (сервисы). Пример микросервисной архитектуры: сервисы для разных бизнес-функций (платежи, пользователи и т.д.), где каждый компонент обрабатывает свою логику. Преимущества микросервисной архитектуры:

1. Несколько резервных копий сервисов;
2. Отсутствие единой точки отказа.

## 4 Примеры распределённых систем.

1. Распределённая система компьютеров — компьютерная сеть;
2. Распределённая система управления — система управления технологическим процессом;
3. Распределённая файловая система — сетевые файловые системы;
4. Распределённая система доменных имён — система для получения информации о доменах.

## 5 Интерфейс в распределённых системах.

Интерфейс — общая граница, через которую передаётся информация (стандарт ISO 24765). В вычислительной системе взаимодействие может осуществляться на пользовательском, программном и аппаратном уровнях.

## 6 Способ взаимодействия физических устройств в распределённых системах.

Физический (аппаратный) интерфейс — это способ взаимодействия устройств с системой через порты (разъёмы). Пример физического интерфейса: подключение клавиатуры к компьютеру. Сетевой интерфейс (шлюз) — это устройство, соединяющее локальную сеть с глобальной (например, Интернет). Стандартный интерфейс — это набор унифицированных технических, программных и конструктивных решений, основанных на стандарте, обеспечивающих взаимодействие элементов системы, их совместимость (информационную, электрическую, конструктивную). Стык — это место соединения сетевых устройств для передачи данных. Связь между протоколами и интерфейсами не всегда однозначна и вот почему:

1. Интерфейс может содержать элементы протокола;
2. Протокол может охватывать несколько интерфейсов (стыков).

Применение стандартных интерфейсов и протоколов:

1. Унификация межсистемных, внутрисистемных, межсетевых и внутрисетевых связей;

2. Повышение эффективности проектирования вычислительных систем.

## 7 Способ взаимодействия программных компонентов.

API интерфейс (Application Programming Interface, Прикладной Программный Интерфейс) – это набор готовых функций, позволяющих программистам взаимодействовать с другими программами. ООП-интерфейс (Объектно-Ориентированного Программирования) – это описание взаимодействия объектов приложения в коде. Голосовые команды в мобильных приложениях или веб-браузерах информационных систем:

1. Добавление к аудиозаписи идентификаторов и метаданных;
2. Передача распознанной команды в интеграционную шину;
3. Получение из шины идентификатора и параметров команды;
4. Отправка и выполнение команды на стороне информационной системы.

Графический веб-интерфейс на основе карты, позволяющий визуализировать прием, обработку, регистрацию и передачу данных. Используется для предоставления цифровых сервисов. Пример использования графического веб-интерфейса: информационное обеспечение и взаимодействие судов и береговых систем мониторинга и управления.

## 8 Способ взаимодействия человека и техники.

Человеко-компьютерное взаимодействие (HCI – Human-Computer Interaction ЧКИ) – это наука о взаимодействии человека и компьютера. Цель ЧКИ – улучшить разработку, оценку и использование интерактивных систем. Интерфейс пользователя – это набор инструментов для взаимодействия с программами и устройствами:

1. Командная строка: команды вводятся с клавиатуры;
2. Графический интерфейс: элементы управления представлены на экране;
3. SILK-интерфейс (Речь-Образ-Язык-Знания, Speech-Image-Language-

Knowledge): управление с помощью речи;

4. Жестовый интерфейс: управление с помощью сенсорного экрана, руля, джойстика и т.п.

5. Нейрокомпьютерный интерфейс: обмен данными между нейронами и компьютером через специальные имплантированные электроды.

## 9 Драйвер.

Драйвер – это программа, позволяющая другим программам (например, операционной системе) работать с аппаратным обеспечением устройств. Операционные системы обычно содержат драйверы для ключевых аппаратных компонентов, без которых система не работает. Другие устройства (видеокарты, принтеры) требуют специальных драйверов от производителя. Драйвер не всегда напрямую управляет устройствами. Он может имитировать их работу (например, драйвер принтера, сохраняющий вывод в файл). Функции драйверов:

1. Предоставление программных сервисов (не связанных с управлением устройствами).
2. Выполнение нулевых операций (`/dev/zero` в Unix).
3. Игнорирование запросов (`/dev/null` в Unix, `NULL` в DOS/Windows).

## 10 Подход к построению драйверов.

Операционная система взаимодействует с виртуальным устройством, понимающим набор команд. Драйвер переводит эти команды в язык, понятный устройству. Это называется абстрагированием от аппаратного обеспечения. В отечественной вычислительной технике такой подход впервые появился в серии ЕС ЭВМ. Управляющее ПО тогда называлось каналным. Драйвер состоит из функций, обрабатывающих события ОС:

1. Загрузка: регистрация в системе, инициализация;
2. Выгрузка: освобождение ресурсов (память, файлы, устройства);
3. Открытие: драйвер открывается программой как файл (`fopen()` в UNIX, `CreateFile()` в Win32);
4. Чтение/запись: обмен данными с устройством;

5. Заккрытие: освобождение ресурсов, уничтожение дескриптора файла;

6. Управление вводом-выводом (IO Control, IOCTL).

Некоторые драйверы поддерживают интерфейс ввода-вывода, специфичный для данного устройства. С помощью этого интерфейса программа имеет возможность отправить специальную команду, которую поддерживает данное устройство. Например, для SCSI-устройств можно отправить команду GETINQUIRY, чтобы получить описание устройства. В Win32-системах

## 11 Разработка драйвера.

Особенность драйверов в Windows: они не обязаны взаимодействовать с внешними устройствами. Это открывает возможности для создания псевдодрайверов, которые расширяют функциональность программ. Пример использования псевдодрайверов:

1. Игровые анти-читы: используют драйверы для наблюдения за всеми процессами в системе;
2. Вредоносное ПО: использует драйверы для обхода защиты.

## 12 Интеграция драйверов.

С развитием 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 платформа. Сначала производители платформ поставляли набор отдельных драйверов для операционных систем, собранный на один носитель, например, компакт-диск, после этого появились установочные пакеты, называвшиеся 4-in-1 и One touch позволявшие упростить установку драйверов в систему. При этом, можно выбрать либо полностью автоматическую установку всех драйверов, либо выбрать вручную нужные. Современный термин — Board Support Package, Пакет поддержки платформы, описывающий такие наборы драйверов устройств. Помимо драйверов, он может содержать модули операционной системы и

программы.

## 13 Примеры интерфейсов в программировании.

### 1. API (Интерфейс программирования приложений):

1.1. Набор готовых инструментов (классы, функции, структуры) для использования в других программах;

1.2. Определяет доступные функции, скрывая детали реализации;

1.3. Обеспечивает взаимодействие между программными компонентами;

1.4. Пример: библиотека функций с описанием сигнатур и семантики.

### 2. COM (Объектная модель компонентов):

2.1. Технология Microsoft для создания ПО из распределенных компонентов;

2.2. Поддерживает полиморфизм и инкапсуляцию ООП;

2.3. Широко используется в Windows (OLE Automation, ActiveX, DCOM, COM+, DirectX);

2.4. Разработан в 1993 году как основа для OLE;

### 3. OLE (Связывание и внедрение объектов):

3.1. Технология Microsoft для связывания и внедрения объектов в другие документы;

3.2. Позволяет передавать задачи между программами редактирования;

3.3. Используется при работе с составными документами, drag-and-drop и буфером обмена.

### 4. ActiveX: 4.1. Переименованная OLE 2.0 (1996);

4.2. Включает элементы управления, документы и Active Scripting;

4.3. Используется веб-дизайнерами для вставки мультимедиа.

### 5. TCP (Протокол управления передачей):

5.1. Один из основных сетевых протоколов Интернета;

5.2. Обеспечивает надежную передачу данных между программами;

5.3. Контролирует длину, скорость и трафик сообщений;

5.4. Используется в электронной почте и обмене файлами.

### 6. ADO (Объекты данных ActiveX):



6.1. Интерфейс программирования для доступа к данным (MS Access, MS SQL Server);

6.2. Представляет данные из разных источников в объектно-ориентированном виде;

6.3. Обеспечивает доступ к dBASE, Access, Excel, Oracle, Paradox, MS SQL Server, Sybase, текстовым файлам, FoxPro, Active Directory Service, Microsoft Jet, Interbase, Informix, PostgreSQL, MySQL и т.д.

7. .NET Framework:

7.1. Открытая инфраструктура приложения;

7.2. Включает вспомогательные программы, библиотеки кода, язык сценариев и другое ПО;

7.3. Облегчает разработку и объединение разных компонентов;

7.4. Обеспечивает совместимость служб, написанных на разных языках;

7.5. Поддерживает создание и выполнение нового поколения приложений и веб-служб XML.