

Многослойная структура ОС

лекция 14

Вычислительную систему, работающую под управлением ОС на основе ядра, можно рассматривать как систему, состоящую из трех иерархически расположенных слоев: нижний слой образует аппаратура, промежуточный — ядро, а утилиты, обрабатывающие программы и приложения, составляют верхний слой системы .

Слоистую структуру вычислительной системы принято изображать в виде системы концентрических окружностей, иллюстрируя тот факт, что каждый слой может взаимодействовать только со смежными слоями. Действительно, при такой организации ОС приложения не могут непосредственно взаимодействовать с аппаратурой, а только через слой ядра.



Многослойный подход является универсальным и эффективным способом декомпозиции сложных систем любого типа, в том числе и программных.

В соответствии с этим подходом система состоит из иерархии слоев.

Каждый слой обслуживает вышележащий слой, выполняя для него некоторый набор функций, которые образуют межслойный интерфейс

На основе функций нижележащего слоя следующий (вверх по иерархии) слой строит свои функции — более сложные и более мощные, которые, в свою очередь, оказываются примитивами для создания еще более мощных функций вышележащего слоя.

Строгие правила касаются только взаимодействия между слоями системы, а между модулями внутри слоя связи могут быть произвольными.

Отдельный модуль может выполнить свою работу либо самостоятельно, либо обратиться к другому модулю своего слоя, либо обратиться за помощью к нижележащему слою через межслойный интерфейс.

Такая организация системы имеет много достоинств. Она существенно упрощает разработку системы, так как позволяет сначала определить «сверху вниз» функции слоев и межслойные интерфейсы, а затем при детальной реализации постепенно наращивать мощность функций слоев, двигаясь «снизу вверх»..

Кроме того, при модернизации системы можно изменять модули внутри слоя без необходимости производить какие-либо изменения в остальных слоях, если при этих внутренних изменениях межслойные интерфейсы остаются в силе

Поскольку ядро представляет собой сложный многофункциональный комплекс, то многослойный подход обычно распространяется и на структуру ядра.

Ядро может состоять из следующих слоев:

- **Средства аппаратной поддержки ОС.**

До сих пор об операционной системе говорилось как о комплексе программ, но, вообще говоря, часть функций ОС может выполняться и аппаратными средствами.

Поэтому иногда можно встретить определение операционной системы как совокупности программных и аппаратных средств.

К операционной системе относят, естественно, не все аппаратные устройства компьютера, а только средства аппаратной поддержки ОС, то есть те, которые прямо участвуют в организации вычислительных процессов: средства поддержки привилегированного режима, систему прерываний, средства переключения контекстов процессов, средства защиты областей памяти и т. п.

- **Машинно-зависимые компоненты ОС.**

Этот слой образуют программные модули, в которых отражается специфика аппаратной платформы компьютера. В идеале этот слой полностью экранирует вышележащие слои ядра от особенностей аппаратуры. Это позволяет разрабатывать вышележащие слои на основе машинно-независимых модулей, существующих в единственном экземпляре для всех типов аппаратных платформ, поддерживаемых данной ОС. Примером экранирующего слоя может служить слой HAL операционной системы Windows NT

- **Базовые механизмы ядра.**

Этот слой выполняет наиболее примитивные операции ядра, такие как программное переключение контекстов процессов, диспетчеризацию прерываний, перемещение страниц из памяти на диск и обратно и т. п.

Модули данного слоя не принимают решений о распределении ресурсов — они только обрабатывают принятые «наверху» решения, что и дает повод называть их исполнительными механизмами для модулей верхних слоев.

Например, решение о том, что в данный момент нужно прервать выполнение текущего процесса А и начать выполнение процесса В, принимается менеджером процессов на вышележащем слое, а слою базовых механизмов передается только директива о том, что нужно выполнить переключение с контекста текущего процесса на контекст процесса В.

- **Менеджеры ресурсов.**

Этот слой состоит из мощных функциональных модулей, реализующих стратегические задачи по управлению основными ресурсами вычислительной системы.

Обычно на данном слое работают менеджеры (называемые также диспетчерами) процессов, ввода-вывода, файловой системы и оперативной памяти.

Разбиение на менеджеры может быть и несколько иным, например менеджер файловой системы иногда объединяют с менеджером ввода-вывода, а функции управления доступом пользователей к системе в целом и ее отдельным объектам поручают отдельному менеджеру безопасности.

Каждый из менеджеров ведет учет свободных и используемых ресурсов определенного типа и планирует их распределение в соответствии с запросами приложений.

Например, менеджер виртуальной памяти управляет перемещением страниц из оперативной памяти на диск и обратно.

Менеджер должен отслеживать интенсивность обращений к страницам, время пребывания их в памяти, состояния процессов, использующих данные, и многие другие параметры, на основании которых он время от времени принимает решения о том, какие страницы необходимо выгрузить и какие — загрузить.

Для исполнения принятых решений менеджер обращается к нижележащему слою базовых механизмов с запросами о загрузке (выгрузке) конкретных страниц.

Внутри слоя менеджеров существуют тесные взаимные связи, отражающие тот факт, что для выполнения процессу нужен доступ одновременно к нескольким ресурсам — процессору, области памяти, возможно, к определенному файлу или устройству ввода-вывода.

Например, при создании процесса менеджер процессов обращается к менеджеру памяти, который должен выделить процессу определенную область памяти для его кодов и данных.

- **Интерфейс системных вызовов.**

Этот слой является самым верхним слоем ядра и взаимодействует непосредственно с приложениями и системными утилитами, образуя прикладной программный интерфейс операционной системы.

Функции API, обслуживающие системные вызовы, предоставляют доступ к ресурсам системы в удобной и компактной форме, без указания деталей их физического расположения.

Например, в операционной системе UNIX с помощью системного вызова `fd = open("/doc/a.txt", O_RDONLY)` приложение открывает файл `a.txt`, хранящийся в каталоге `/doc`, а с помощью системного вызова `read(fd, buffer, count)` читает из этого файла в область своего адресного пространства, имеющую имя `buffer`, некоторое количество байт.

Для осуществления таких комплексных действий системные вызовы обычно обращаются за помощью к функциям слоя менеджеров ресурсов, причем для выполнения одного системного вызова может понадобиться несколько таких обращений.



Приведенное разбиение ядра ОС на слои является достаточно условным. В реальной системе количество слоев и распределение функций между ними может быть и иным. В системах, предназначенных для аппаратных платформ одного типа, например ОС NetWare, слой машинно-зависимых модулей обычно не выделяется, сливаясь со слоем базовых механизмов и, частично, со слоем менеджеров ресурсов.

Возможна и противоположная картина, когда ядро состоит из большего количества слоев. Например, менеджеры ресурсов, составляя определенный слой ядра, в свою очередь, могут обладать многослойной структурой. Прежде всего это относится к менеджеру ввода-вывода, нижний слой которого составляют драйверы устройств, например драйвер жесткого диска или драйвер сетевого адаптера, а верхние слои — драйверы файловых систем или протоколов сетевых служб, имеющие дело с логической организацией информации.

Способ взаимодействия слоев в реальной ОС также может отклоняться от описанной выше схемы.

Для ускорения работы ядра в некоторых случаях происходит непосредственное обращение с верхнего слоя к функциям нижних слоев, минуя промежуточные.

Типичным примером такого «неправильного» взаимодействия является начальная стадия обработки системного вызова.

обращаясь прямо к базовым механизмам ядра.

На многих аппаратных платформах для реализации системного вызова используется инструкция программного прерывания.

Этим приложение фактически вызывает модуль первичной обработки прерываний, который находится в слое базовых механизмов, а уже этот модуль вызывает нужную функцию из слоя системных вызовов.

Сами функции системных вызовов также иногда нарушают субординацию иерархических слоев,

Выбор количества слоев ядра является ответственным и сложным делом: увеличение числа слоев ведет к некоторому замедлению работы ядра за счет дополнительных накладных расходов на межслойное взаимодействие, а уменьшение числа слоев ухудшает расширяемость и логичность системы.

Обычно операционные системы, прошедшие долгий путь эволюционного развития, например многие версии UNIX, имеют неупорядоченное ядро с небольшим числом четко выделенных слоев, а у сравнительно «молодых» операционных систем, таких как Windows NT, ядро разделено на большее число слоев и их взаимодействие формализовано в гораздо большей степени.