

Дисциплина: «Теория операционных систем»

Преподаватель:
Алексеев Кирилл Анатольевич

Тема 1

Архитектура, назначение и функции операционных систем

практическое занятие

занятие 1.2

Архитектура операционной системы

Учебные вопросы:

- 1) Архитектура операционной системы.
- 2) Классификация операционных систем.
- 3) Требования, предъявляемые к ОС.
- 4) Множественные прикладные среды и виртуальные машины.

Литература

Основная:

- 1) Назаров С. В., Гудыно Л. П., Кириченко А. А. Операционные системы. Практикум.
- 2) Олифер В.Г., Олифер Н.А. Сетевые операционные системы.
- 3) Таненбаум Э. Современные операционные системы.

Дополнительная:

- 1) Староверова Н. А. Операционные системы.
- 2) Стащук П. В. Краткое введение в операционные системы.



Архитектура операционной системы - структурная и функциональная организация ОС на основе некоторой совокупности программных модулей.

В состав ОС входят исполняемые и объектные модули стандартных для данной ОС форматов, программные модули специального формата (например, загрузчик ОС, драйверы ввода-вывода), конфигурационные файлы, файлы документации, модули справочной системы и т.д.

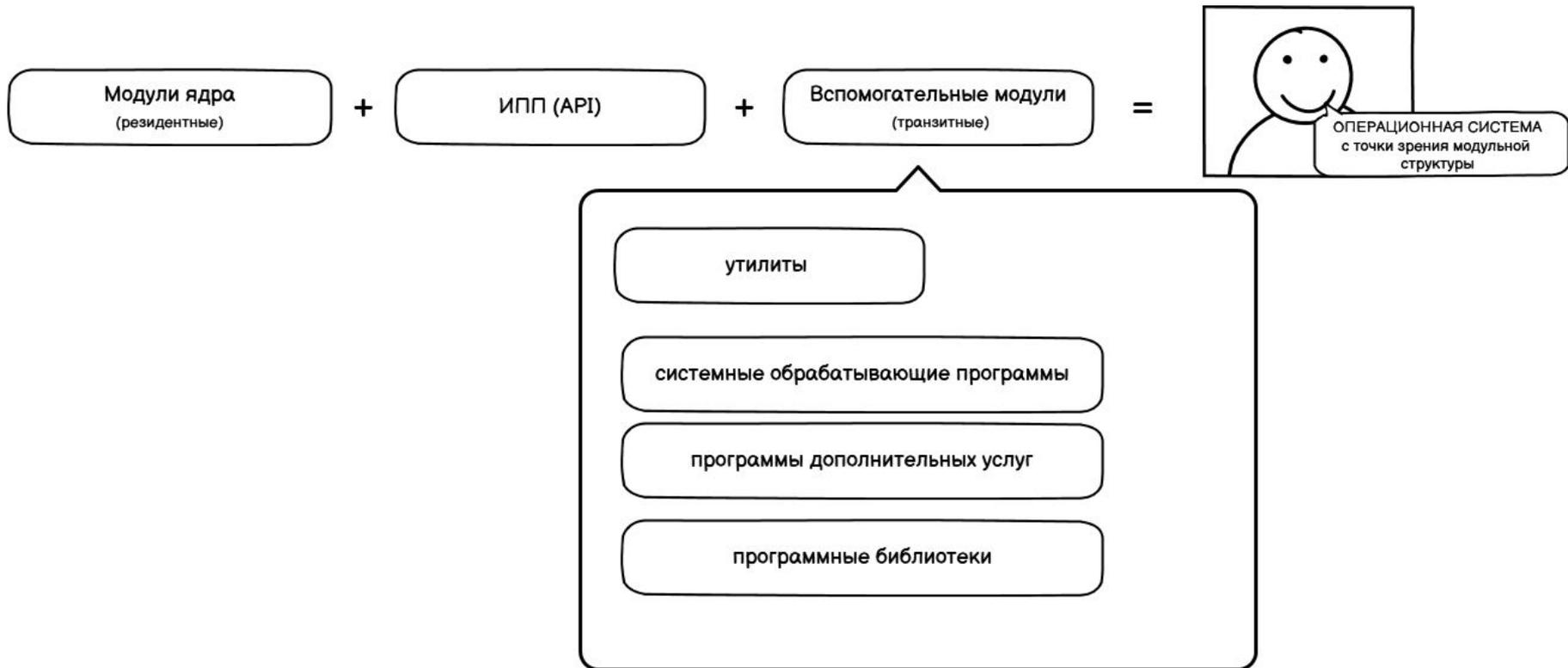
Классической считается архитектура ОС, основанная на концепции **иерархической многоуровневой машины, привилегированном ядре (состоящем из модулей ядра) и пользовательском режиме работы транзитных модулей.**



Модули ядра выполняют базовые функции ОС: управление процессами, памятью, устройствами ввода-вывода и т.п.

Модули ОС, оформленные в виде утилит, системных обрабатывающих программ и библиотек, загружаемые в оперативную память только на время выполнения своих функций, называются **транзитными**.

Особый класс функций ядра служит для поддержки приложений, создавая для них **прикладную программную среду**. Функции ядра, которые могут вызываться приложениями, образуют **интерфейс прикладного программирования – API (Application Programming Interface)**.



Многослойная структура ОС



1. Средства аппаратной поддержки ОС. К ним относятся: система прерываний, средства поддержки привилегированного режима, средства поддержки виртуальной памяти, системный таймер, средства переключения контекстов процессов (информация о состоянии процесса в момент его приостановки), средства защиты памяти и другие.

2. Машинно-зависимые модули ОС. Этот слой образует модули, в которых отражается специфика аппаратной платформы компьютера. Назначение этого слоя – "экранирование" вышележащих слоев ОС от особенностей аппаратуры, (*уровень аппаратных абстракций*).

3. Базовые механизмы ядра. Этот слой модулей выполняет наиболее примитивные операции ядра: программное переключение контекстов процессов, диспетчеризацию прерываний, перемещение страниц между основной памятью и диском и т.п.



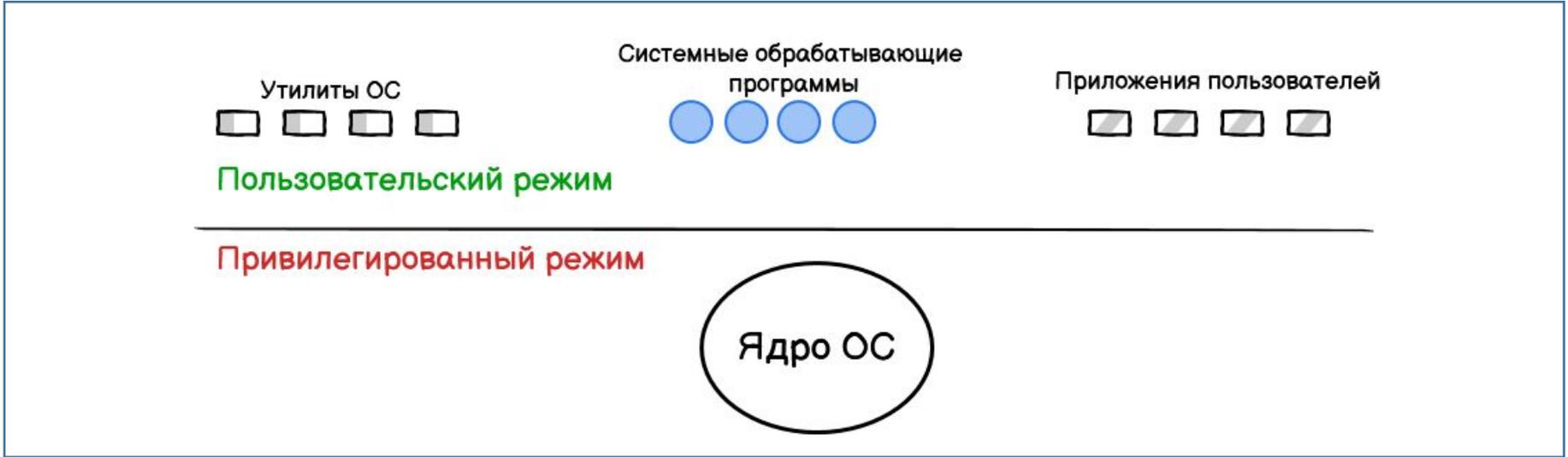
4. Менеджеры ресурсов. Модули этого слоя выполняют стратегические задачи по управлению ресурсами вычислительной системы.

5. Интерфейс системных вызовов. Это верхний слой ядра ОС, взаимодействующий с приложениями и системными утилитами, он образует прикладной программный интерфейс ОС.



Аппаратура компьютера должна поддерживать как минимум два режима работы — **пользовательский** (user mode) и **привилегированный**, который также называют **режимом ядра** (kernel mode), или **супервизора** (supervisor mode).

Подразумевается, что операционная система или некоторые ее части работают в привилегированном режиме, а приложения — в пользовательском режиме.



Обработка системного вызова

Кл



Признак	Привилег. режим	Польз. режим
Возможность выполнения машинных инструкций переключения процессора с задачи на задачу	Есть	Нет
Возможность управления устройствами ввода-вывода, системой прерываний	Есть	Нет
Доступ ко всем областям памяти, работа с физической памятью	Есть	Нет
Доступ процесса к виртуальной памяти этого же процесса	Есть	Есть
Возможность управления доступом к механизмам распределения и защиты памяти	Есть	Нет



Виды ядер современных ОС:

- 1. Наноядро.** Крайне упрощённое и минимальное ядро, выполняет лишь одну задачу – обработку аппаратных прерываний, генерируемых устройствами компьютера. После обработки посылает информацию о результатах обработки вышележащему программному обеспечению. НЯ используются для виртуализации аппаратного обеспечения реальных компьютеров или для реализации механизма гипервизора.
- 2. Микроядро** предоставляет только элементарные функции управления процессами и минимальный набор абстракций для работы с оборудованием. Большая часть работы осуществляется с помощью специальных пользовательских процессов, называемых сервисами. В микроядерной операционной системе можно, не прерывая ее работы, загружать и выгружать новые драйверы, файловые системы и т. д. Микроядерными являются ядра ОС Minix и GNU Hurd и ядро систем семейства BSD. Классическим примером микроядерной системы является Symbian OS.

Виды ядер современных ОС:

3. Экзоядро предоставляет лишь набор сервисов для взаимодействия между приложениями, а также необходимый минимум функций, связанных с защитой: выделение и высвобождение ресурсов, контроль прав доступа и т. д. ЭЯ не занимается предоставлением абстракций для физических ресурсов – эти функции выносятся в библиотеку пользовательского уровня (так называемую libOS). В отличие от микроядра ОС, базирующиеся на ЭЯ, обеспечивают большую эффективность за счет отсутствия необходимости в переключении между процессами при каждом обращении к оборудованию.



Виды ядер современных ОС:

4. Монолитное ядро предоставляет широкий набор абстракций оборудования. Все части ядра работают в одном адресном пространстве. МНЯ требуют перекомпиляции при изменении состава оборудования. Компоненты операционной системы являются не самостоятельными модулями, а составными частями одной программы. МНЯ более производительны, чем микроядро, поскольку работают как один большой процесс. МНЯ является большинство Unix-систем и Linux. Монолитность ядер усложняет отладку, понимание кода ядра, добавление новых функций и возможностей, удаление ненужного, унаследованного от предыдущих версий кода.



Виды ядер современных ОС:

5. Модульное ядро – современная, усовершенствованная модификация архитектуры МЯ. В отличие от "классических" МНЯ, модульные ядра не требуют полной перекомпиляции ядра при изменении состава аппаратного обеспечения компьютера. Вместо этого они предоставляют тот или иной механизм подгрузки модулей, поддерживающих то или иное аппаратное обеспечение (например, драйверов). Подгрузка модулей может быть как динамической, так и статической (при перезагрузке ОС после переконфигурирования системы). Мод. Я удобнее для разработки, чем традиционные монолитные ядра. Они предоставляют программный интерфейс (API) для связывания модулей с ядром, для обеспечения динамической подгрузки и выгрузки модулей. Не все части ядра могут быть сделаны модулями. Некоторые части ядра всегда обязаны присутствовать в оперативной памяти и должны быть жёстко "вшиты" в ядро.

Виды ядер современных ОС:

6. Гибридное ядро – модифицированные микроядра, позволяющие для ускорения работы запускать "несущественные" части в пространстве ядра. Имеют "гибридные" достоинства и недостатки. Примером смешанного подхода может служить возможность запуска операционной системы с монолитным ядром под управлением микроядра. Так устроены 4.4 BSD и MkLinux, основанные на микроядре Mach. Микроядро обеспечивает управление виртуальной памятью и работу низкоуровневых драйверов. Все остальные функции, в том числе взаимодействие с прикладными программами, осуществляются монолитным ядром. Данный подход сформировался в результате попыток использовать преимущества микроядерной архитектуры, сохраняя по возможности хорошо отлаженный код монолитного ядра.

По назначению

- универсальные
- специализированные

По способу загрузки

- загружаемые ОС
- системы, постоянно находящиеся в памяти вычислительной системы

По аппаратной платформе (типу вычислительной техники)

- операционные системы для смарт-карт
- встроенные операционные системы
- операционные системы для персональных компьютеров
- операционные системы мэйнфреймов
- серверные операционные системы
- кластерные операционные системы

По особенностям алгоритмов управления ресурсами

- Поддержка многозадачности (многопрограммности):
 - однопрограммные (однозадачные)
 - многопрограммные (многозадачные)
- Поддержка многопользовательского режима:
 - однопользовательские
 - многопользовательские
- Виды многопрограммной работы:
 - не вытесняющая многопрограммность (процесс сам отдает управление ОС)
 - вытесняющая многопрограммность (решение о переключении процессов принимает ОС)
- Многопроцессорная обработка:
 - асимметричные ОС (выполняются на одном процессоре, распределяя прикладные задачи по остальным процессорам)
 - симметричные ОС (децентрализованная система)

По области использования и форме эксплуатации (по типу обработки данных)

- системы пакетной обработки (для решения задач вычислительного характера, не требующих быстрого получения результатов)
- системы разделения времени (удобство и эффективность работы пользователя, который имеет терминал и может вести диалог со своей программой)
- системы реального времени (предназначены для управления техническими объектами (станок, спутник, технологический процесс, например доменный и т.п.), где существует предельное время на выполнение программ, управляющих объектом)

По способу загрузки

- загружаемые ОС
- системы, постоянно находящиеся в памяти вычислительной системы

По аппаратной платформе (типу вычислительной техники)

- операционные системы для смарт-карт
- встроенные операционные системы
- операционные системы для персональных компьютеров
- операционные системы мэйнфреймов
- серверные операционные системы
- кластерные операционные системы



1) **Функциональной полноты**

Современная ОС должна поддерживать мультипрограммную обработку, виртуальную память, свопинг, развитый интерфейс пользователя (многооконный графический, аудио -, менюориентированный и т.д.).

2) **Эффективность** (степень соответствия системы своему назначению, которая оценивается некоторым множеством показателей эффективности).

Часто эффективность ОС оценивают ее производительностью (пропускной способностью) – количеством задач пользователей, выполняемых за некоторый промежуток времени, временем реакции на запрос пользователя и др.

3) **Надежность и отказоустойчивость.** Система должна быть защищена как от внутренних, так и от внешних сбоев и отказов. В случае ошибки в программе или аппаратуре система должна обнаружить ошибку и попытаться исправить положение или постараться свести к минимуму ущерб, нанесенный этой ошибкой.



- 4) Безопасность (защищенность).** В ОС должны быть средства аутентификации – определения легальности пользователей, авторизации – предоставления легальным пользователям установленных им прав доступа к ресурсам, аудита – фиксации всех потенциально опасных для системы событий, а также средства обеспечения безопасности сервисов ОС.
- 5) Расширяемость.** Если программный код модулей ОС написан таким образом, что дополнения и изменения могут вноситься без нарушения целостности системы, то такую ОС называют расширяемой.
- 6) Переносимость.** Код ОС должен максимально легко переноситься с аппаратной платформы (которые различаются не только типом процессора, но и способом организации всей аппаратуры компьютера) одного типа на аппаратную платформу другого типа.



7) Совместимость. Если ОС имеет средства для выполнения прикладных программ, написанных для других операционных систем, то она совместима с этими системами. Следует различать совместимость на уровне двоичных кодов и совместимость на уровне исходных текстов.

8) Удобство. Простота и гибкость средств ОС, обеспечение пользователю максимально возможного удобства при работе с ОС (наличие графического интерфейса, документации, средств настройки ОС).

9) Масштабируемость. Заключается в управлении вычислительной системы с различным числом процессоров. При этом предусматривается линейное возрастание производительности или резервирование аппаратных средств.