

Состав компонентов ОС

Состав компонентов ОС

1. Управление процессами
2. Управление памятью
3. Управление файлами
4. Управление внешними устройствами
5. Защита данных
6. Администрирование
7. Интерфейс прикладного программирования
8. Пользовательский интерфейс

Состав компонентов ОС

6. Администрирование
7. Интерфейс прикладного программирования
8. Пользовательский интерфейс

Архитектура ОС

Архитектура ОС – это базовая организация системы, воплощенная в ее компонентах, их отношениях между собой и с окружением, а также принципы, определяющие проектирование и развитие системы (стандарт института инженеров по электротехнике и электронике, IEEE 1471).

Архитектура ОС – это структурная и функциональная организация ОС.

Принципы разработки современных ОС

1. Концепция многоуровневой иерархической вычислительной системы (виртуальной машины) с ОС многослойной структуры.
2. Разделение модулей ОС по функциям на две группы:
 - ✓ ядро – модули, выполняющие основные функции ОС,
 - ✓ модули, выполняющие остальные (вспомогательные) функции.

Ядро – это модуль, который выполняется в режиме ядра (привилегированном режиме), в котором доступно все, все ресурсы, пользовательские приложения. Ядро выполняет основные функции.

Принципы разработки современных ОС

3.Разделение модулей ОС по размещению в памяти вычислительной системы:

✓ **резидентные** — постоянно находящиеся в оперативной памяти,

✓ **транзитные** — загружаемые в оперативную память только на время выполнения своих функций.

Принципы разработки современных ОС

4. Реализация двух режимов работы вычислительной системы:

- ✓ **привилегированного режима** (режима ядра);
- ✓ **пользовательского режима.**

5. Ограничение функций ядра (числа его модулей) до минимально необходимых функций.

Принципы разработки современных ОС

6. Модульное строение

а) однократно используемые модули – используются только при загрузке ОС;

б) повторно используемые модули:

✓ **привилегированные модули** – не допускают исполнение прерываний, при их загрузке должны проработать до конца;

✓ **реентерабельные** допускают повторный запуск и допускают прерывания в любом месте модуля;

✓ **повторно входимые** – допускают прерывания после завершения секций.

Принципы разработки современных ОС

7. Параметрическая универсальность. Возможность генерации ОС и создания нескольких рабочих конфигураций.
8. Функциональная избыточность.
9. Функциональная избирательность.

Принципы разработки современных ОС

10.Открытость, модифицируемость, расширяемость
(возможность получения текстов исходных
модулей).

11.Мобильность

12.Совместимость

13.Безопасность

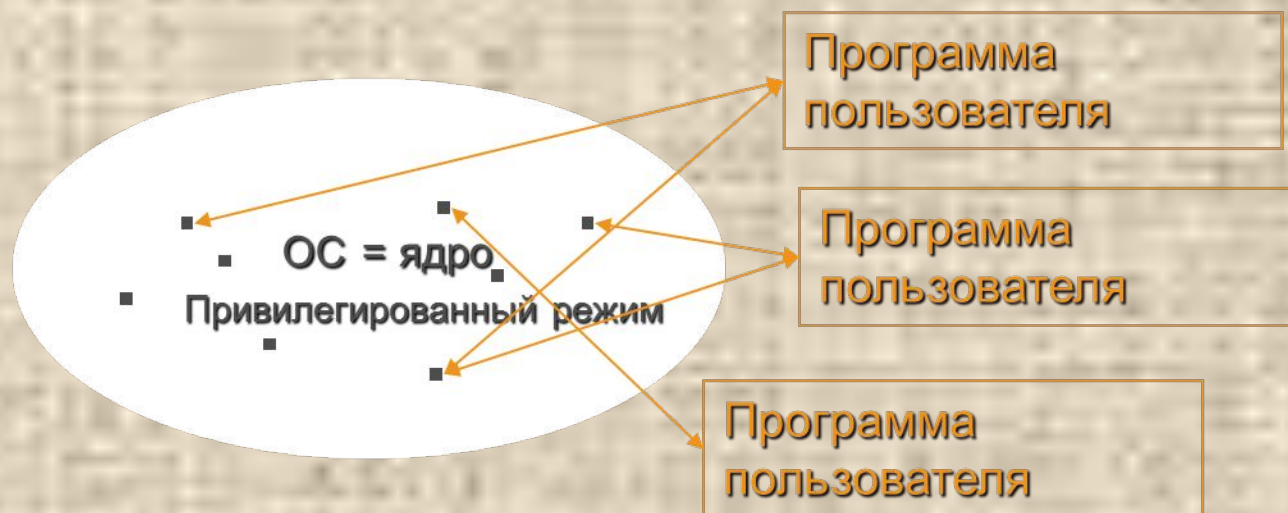
Модульно – интерфейсный подход (структурный подход)

1. Декомпозиция системы на модули по структурному или функциональному признаку.
2. Модули и их взаимные связи образуют абстракцию системы высокого уровня.
3. Описывается каждый модуль и определяется его интерфейс.
4. Проводится декомпозиция каждого модуля и т. д.

Типы архитектур ОС

1. Монолитная архитектура

- - точки входа в ядро – системные вызовы



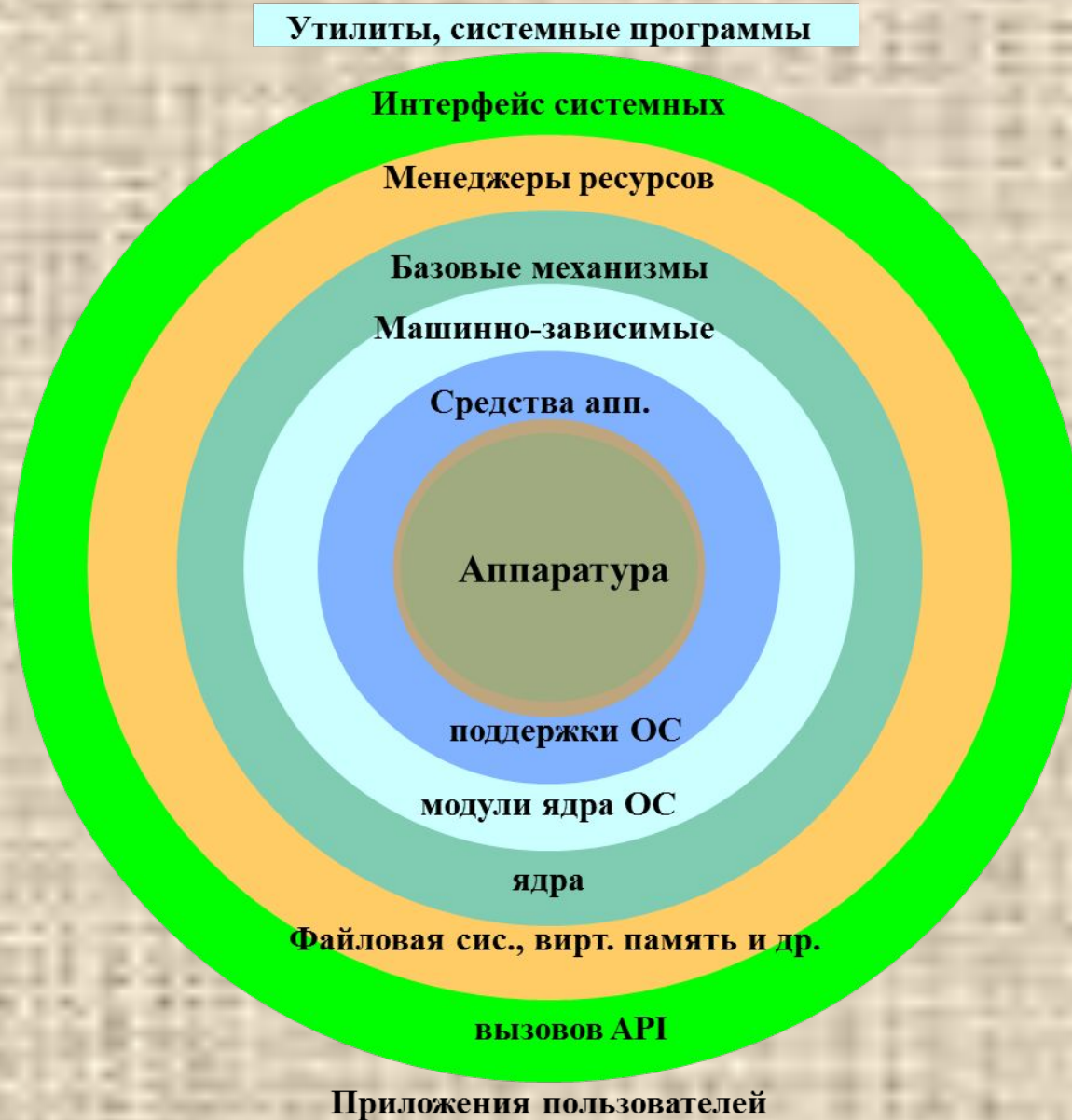
1. Монолитная архитектура

- 1) Каждая процедура может вызывать каждую.
- 2) Все процедуры работают в привилегированном режиме.
- 3) Ядро совпадает со всей операционной системой.
- 4) Пользовательские программы взаимодействуют с ядром через системные вызовы.

2. Многоуровневая архитектура



2. Многоуровневая архитектура



2. Многоуровневая архитектура

- 1) Операционная система представляется в виде иерархии слоев.
- 2) Верхний слой определяет виртуальную машину с желаемыми свойствами.
- 3) Каждый следующий слой детализирует вышележащий, выполняя для него некоторый набор функций.

2. Многоуровневая архитектура

- 4) Межслойные интерфейсы подчиняются строгим правилам. Связи внутри слоя могут быть произвольными.
- 5) Отдельный модуль слоя N может выполнить работу самостоятельно или по следующим вариантам:
 - обратиться только к слою $N-1$;
 - обратиться к некоторой команде слоя N .

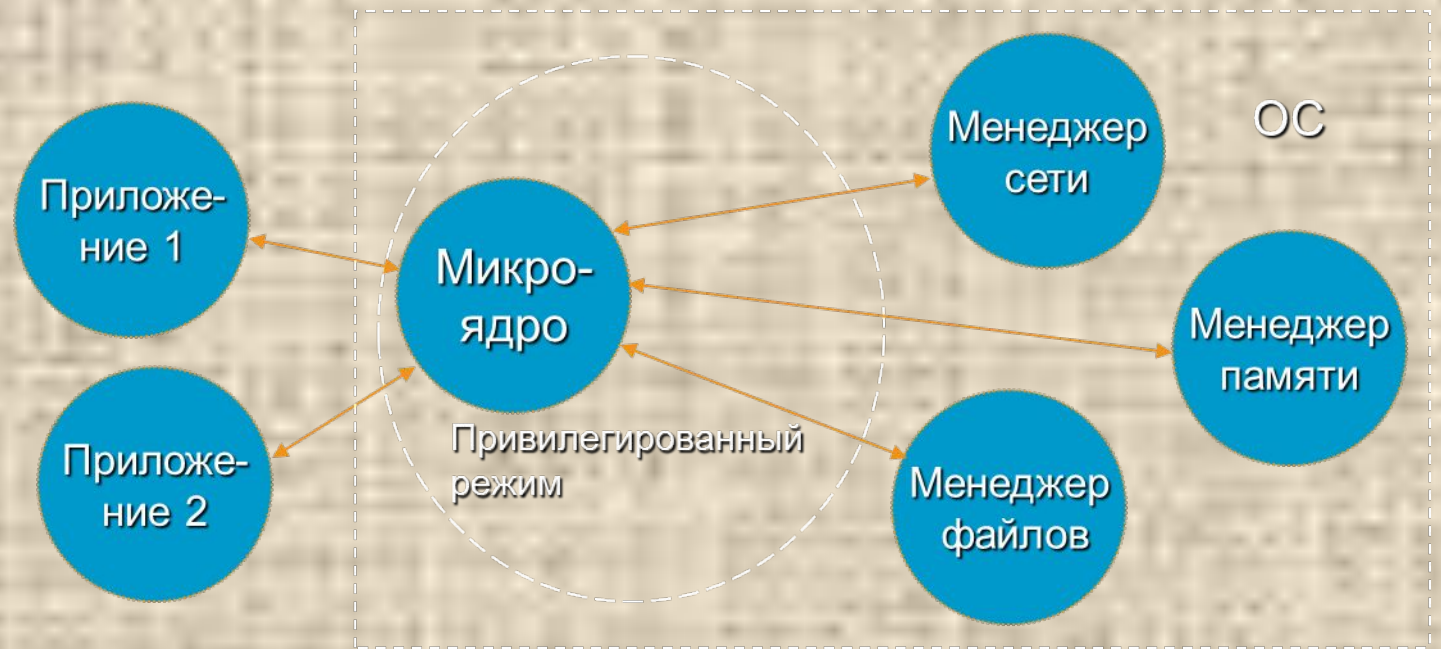
Достоинства:

1. Между уровнями можно организовать четкий интерфейс.
2. Систему можно спроектировать методом «сверху вниз», а реализовать методом «снизу вверх».
3. Уровни реализуются в соответствии с их порядком, начиная с аппаратуры и далее вверх.

Достоинства:

4. Каждую новую виртуальную машину можно детально проверить, после чего продолжать дальнейшую работу.
5. Любой слой достаточно просто модифицировать, не затрагивая другие слои и не меняя межслойные интерфейсы.

3. Микроядерная архитектура



3. Микроядерная архитектура

Ядро минимальных размеров, выполняет только наиболее важные функции.

Функции микроядра:

- ✓ взаимодействие между программами;
- ✓ планирование использования процессора;
- ✓ первичная обработка прерываний и операций ввода-вывода;
- ✓ базовое управление памятью.

Монолитное ядро — необходимость перекомпиляции при каждом изменении, большой объем ядра, сложность отладки, высокая скорость работы.

Многоуровневые системы — необходимость перекомпиляции при изменениях, отлаживается только измененный уровень, меньшая скорость работы.

Микроядро – простота отладки, возможность замены компонент без перекомпиляции и остановки системы, очень медленные.

Наноядро – это уменьшенный вариант микроядра. Современные ОС, как правило, совмещают элементы различных подходов.