

# OS Hard. Теория

Александр Сергеев

## 1 Архитектура операционной системы

Архитектура любой операционной системы разбивается на пространство ядра и пространство пользователя

Обращение приложений пространства ядра к функциям пространства пользователя происходит через *системные вызовы*

В пространстве ядра работает ядро и драйвера

### 1.1 Архитектура ядра

#### 1.1.1 Монолитное ядро

Весь код ядра выполняется в пространстве ядра

При этом оно все еще остается модульным: модули общаются по средствам интерфейсов, они могут быть включены или отключены

#### 1.1.2 Микро-ядро

В пространстве ядра выполняется минимальное количество кода

Многие подсистемы (серверы) вынесены в пространство пользователя

Взаимодействие между ядром и модулями в пространстве пользователя используются в пространстве пользователя, что требует больше времени

Но ошибки в серверах не убивают ядро, в отличие от монолитных ядер

Операционная система Windows заявлена как "гибридная"

Основные компоненты работают в пространстве ядра

На практике "гибридность" доказать сложно из-за закрытости кода

## 1.2 Мульти-процессное взаимодействие

Существует *кооперативное* и *вытесняющее* исполнение

При кооперативном исполнении процесс сам передает управление операционной системе в момент простоя

При вытесняющем исполнении процесс приостанавливается ядром

Единица времени исполнения при вытесняющем исполнении называется *time slice* (к примеру, 100 мс)

Поддержка кооперативного и вытесняющего исполнения должна быть реализована в ядре

Не-вытесняющие ядра существовали до Linux 2.4