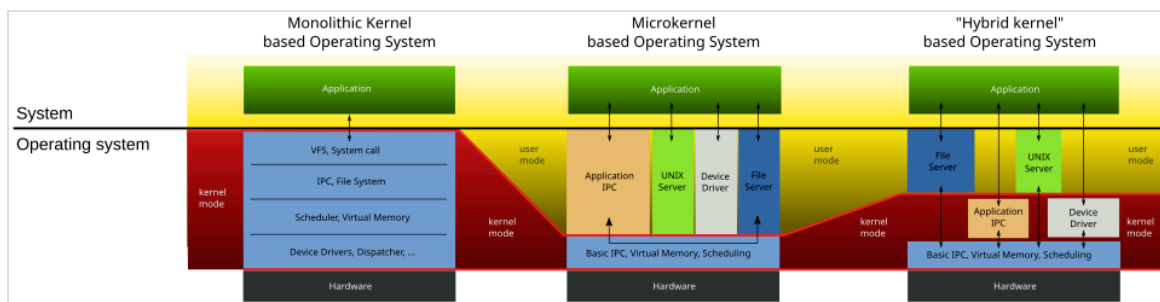




# Монолитное ядро



Структура операционных систем на основе монолитного ядра, микроядра и гибридного ядра

Монолитное **ядро** — это архитектура операционной системы , в которой вся операционная система работает в пространстве ядра . Монолитная модель отличается от других архитектур, таких как микроядро <sup>[ 1 ] [ 2 ]</sup> , тем, что она сама определяет высокоуровневый виртуальный интерфейс для аппаратного обеспечения компьютера . Набор примитивов или системных вызовов реализует все службы операционной системы, такие как управление процессами , параллелизм и управление памятью .

Драйверы устройств могут быть добавлены в ядро как загружаемые модули ядра .

## Примеры

- Большинство ядер BSD
  - FreeBSD
  - OpenBSD
  - NetBSD
- ядро Linux
  - Андроид
- Другие ядра Unix / Unix-подобных систем
  - ЭКС
  - Oracle Solaris
- MS-DOS
  - Windows 9x
- OpenVMS
- Palm OS (версия ≤ 5.0)

## Загружаемые модули

---

Модульные операционные системы, такие как OS-9 и большинство современных операционных систем с монолитным ядром, таких как OpenVMS , Linux , FreeBSD , NetBSD , DragonFly BSD , Solaris и AIX , могут динамически загружать (и выгружать) исполняемые модули ядра во время выполнения.

Эта модульность операционной системы реализуется на уровне двоичных файлов (образов), а не на уровне архитектуры. Модульные монолитные операционные системы не следует путать с архитектурным уровнем модульности, присущим клиент-серверным операционным системам (и их производным, иногда называемым гибридным ядром ), которые используют микроядра и серверы (не путать с модулями или демонами).

На практике динамическая загрузка модулей — это просто более гибкий способ работы с образом операционной системы во время выполнения, по сравнению с перезагрузкой с использованием другого образа операционной системы. Модули позволяют легко расширять возможности операционной системы по мере необходимости. <sup>[ 3 ]</sup> Динамически загружаемые модули требуют небольших накладных расходов по сравнению со встраиванием модуля в образ операционной системы.

Однако в некоторых случаях динамическая загрузка модулей (по мере необходимости) помогает свести к минимуму объём кода, работающего в пространстве ядра ; например, для минимизации нагрузки на операционную систему встраиваемых устройств или устройств с ограниченными аппаратными ресурсами. В частности, незагруженный модуль не нужно хранить в дефицитной оперативной памяти .

## Смотрите также

---

- Экзоядро
- Гибридное ядро
- Ядро (операционная система)
- Микроядро
- Наноядро
- Дебаты Таненбаума и Торвальдса

## Ссылки

---

1. «Программирование модульных систем в Minix3» (<http://www.usenix.org/publications/login/2006-04/openpdfs/herder.pdf>) (PDF) .
2. «Сервер-клиент, или многоуровневая структура» (<https://web.archive.org/web/20100717085530/http://www.gridbus.org/~raj/microkernel/chap2.pdf>)(PDF) . Проектирование микроядра PARAS. Архивировано из исходного файла (<http://www.gridbus.org/~raj/microkernel/chap2.pdf>)(PDF) 17 июля 2010 г. Дата обращения: 15 июля 2009 г. (<https://web.archive.org/web/20100717085530/http://www.gridbus.org/~raj/microkernel/chap2.pdf>) (<http://www.gridbus.org/~raj/microkernel/chap2.pdf>)
3. "Определение ядра" (<http://www.linfo.org/kernel.html>) . (<http://www.linfo.org/kernel.html>)