

本节内容

# 操作系统的 体系结构

王道考研/CSKAOYAN.COM

1

## 知识总览

### 操作系统体系结构

大内核（又名：宏内核/单内核）

微内核

分层结构

模块化

外核



对新知识充满期待

Tips: 一定是**考简单的选择题**，了解各种体系结构的**特性**，了解各自的**优缺点**

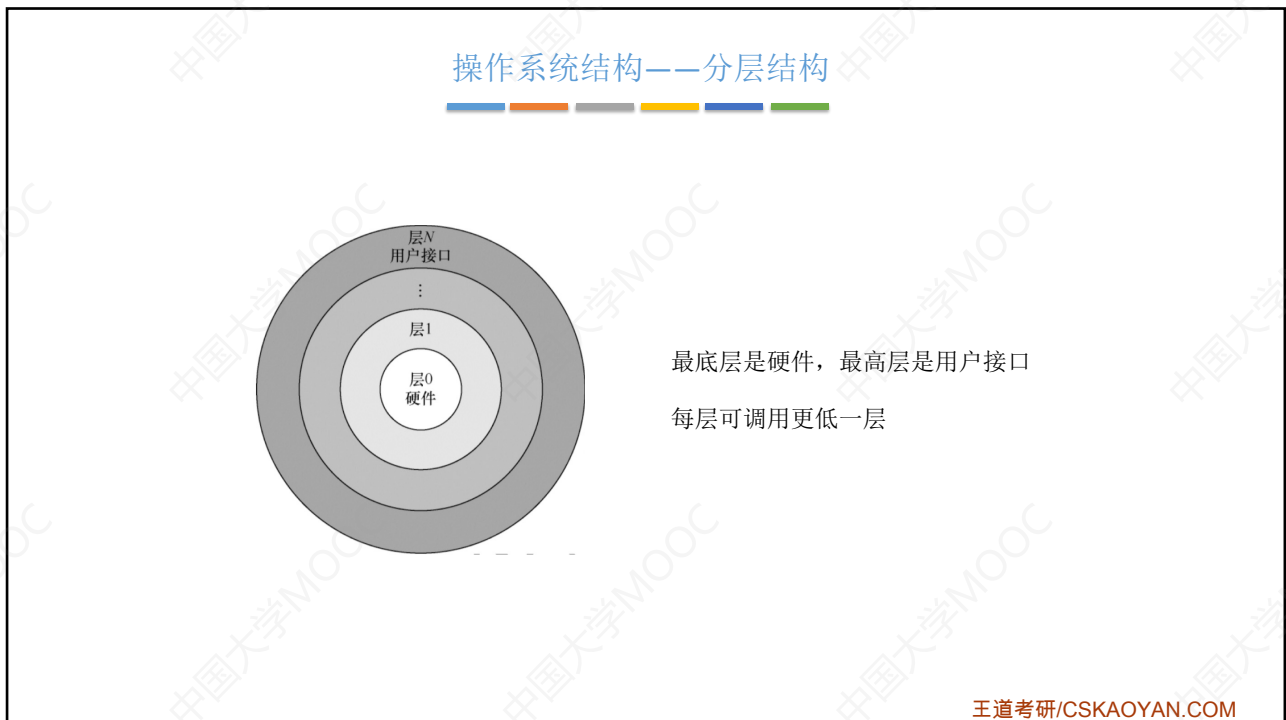
王道考研/CSKAOYAN.COM

2

操作系统结构			
	特性、思想	优点	缺点
分层结构	内核分多层，每层可单向调用更低一层提供的接口	<ul style="list-style-type: none"> <li>1. 便于调试和验证，自底向上逐层调试验证</li> <li>2. 易扩充和易维护，各层之间调用接口清晰固定</li> </ul>	<ul style="list-style-type: none"> <li>1. 仅可调用相邻低层，难以合理定义各层的边界</li> <li>2. 效率低，不可跨层调用，系统调用执行时间长</li> </ul>
模块化	将内核划分为多个模块，各模块之间相互协作。 内核 = 主模块 + 可加载内核模块 主模块：只负责核心功能，如进程调度、内存管理 可加载内核模块：可以动态加载新模块到内核，而无需重新编译整个内核	<ul style="list-style-type: none"> <li>1. 模块间逻辑清晰易于维护，确定模块间接口后即可多模块同时开发</li> <li>2. 支持动态加载新的内核模块（如：安装设备驱动程序、安装新的文件系统模块到内核），增强OS适应性</li> <li>3. 任何模块都可以直接调用其他模块，无需采用消息传递进行通信，效率高</li> </ul>	<ul style="list-style-type: none"> <li>1. 模块间的接口定义未必合理、实用</li> <li>2. 模块间相互依赖，更难调试和验证</li> </ul>
宏内核（大内核）	所有的系统功能都放在内核里（大内核结构的OS通常也采用了“模块化”的设计思想）	<ul style="list-style-type: none"> <li>1. 性能高，内核内部各种功能都可以直接相互调用</li> </ul>	<ul style="list-style-type: none"> <li>1. 内核庞大功能复杂，难以维护</li> <li>2. 大内核中某个功能模块出错，就可能导致整个系统崩溃</li> </ul>
微内核	只把中断、原语、进程通信等最核心的功能放入内核。进程管理、文件管理、设备管理等功能以用户进程的形式运行在用户态	<ul style="list-style-type: none"> <li>1. 内核小功能少、易于维护，内核可靠性高</li> <li>2. 内核外的某个功能模块出错不会导致整个系统崩溃</li> </ul>	<ul style="list-style-type: none"> <li>1. 性能低，需要频繁的切换用户态/核心态。用户态下的各功能模块不可以直接相互调用，只能通过内核的“消息传递”来间接通信</li> <li>2. 用户态下的各功能模块不可以直接相互调用，只能通过内核的“消息传递”来间接通信</li> </ul>
外核（exokernel）	内核负责进程调度、进程通信等功能，外核负责为用户进程分配未经抽象的硬件资源，且由外核负责保证资源使用安全	<ul style="list-style-type: none"> <li>1. 外核可直接给用户进程分配“不虚拟、不抽象”的硬件资源，使用户进程可以更灵活的使用硬件资源</li> <li>2. 减少了虚拟硬件资源的“映射层”，提升效率</li> </ul>	<ul style="list-style-type: none"> <li>1. 降低了系统的一致性</li> <li>2. 使系统变得更复杂</li> </ul>

王道考研/CSKAOYAN.COM

3



王道考研/CSKAOYAN.COM

4

## 操作系统结构——模块化

### (2) 模块化

模块化是将操作系统按功能划分为若干个具有一定独立性的模块。每个模块具有某方面的管理功能，并规定好各模块间的接口，使各模块之间能通过接口进行通信。还可以进一步将各模块细分为若干个具有一定功能的子模块，同样也规定好各子模块之间的接口。把这种设计方法称为模块-接口法，图 1.3 所示为由模块、子模块等组成的模块化操作系统结构。

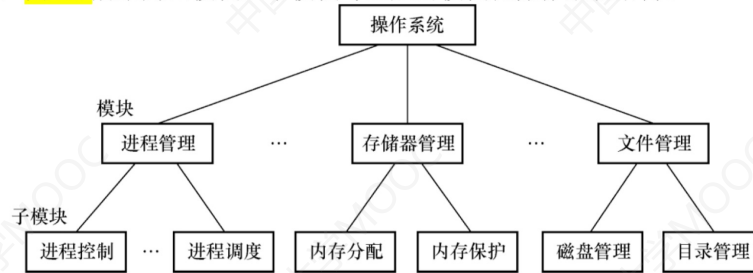
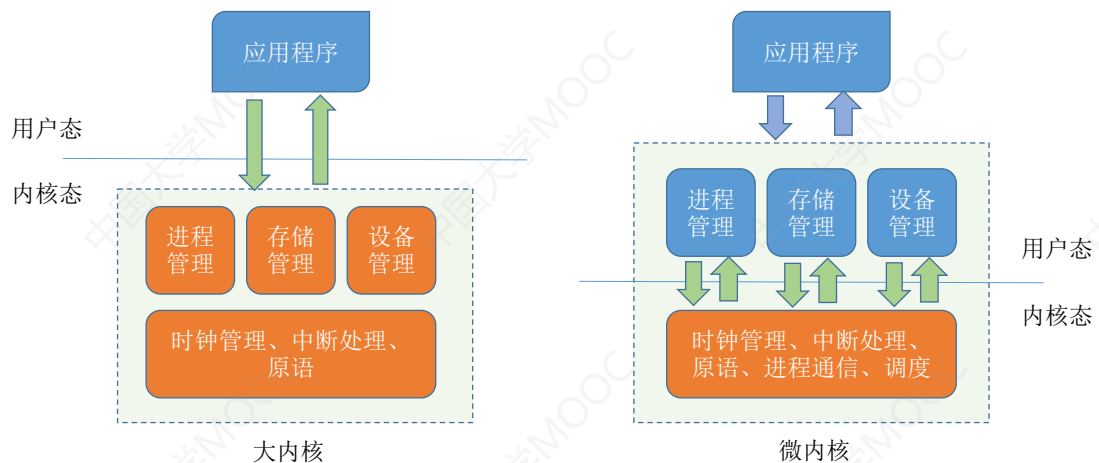


图 1.3 模块化结构的操作系统

王道考研/CSKAOYAN.COM

5

## 操作系统结构——宏内核、微内核



王道考研/CSKAOYAN.COM

6

