

课程说明

王道考研/CSKAOYAN.COM

大家熟悉的操作系统



Mac OS



Windows



ANDROID



王道考研/CSKAOYAN.COM

本节内容

操作系统的 概念（定义） 功能和目标

王道考研/CSKAOYAN.COM

知识总览

```
graph TD; A["操作系统的..."] --> B["概念(定义)"]; A --> C["功能和目标"]; B --> D["什么是操作系统"]; C --> E["操作系统要做些什么？"]
```

什么是操作系统

操作系统要做些什么？

王道考研/CSKAOYAN.COM

操作系统的概念（定义）

一台电脑的诞生~

Step 1: 厂家组装一台裸机

Step 2: 出售前安装操作系统

Step 3: 用户安装应用程序（eg: QQ）

Step 4: 使用 QQ 聊天



王道考研/CSKAOYAN.COM

操作系统的概念（定义）

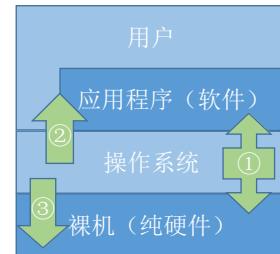
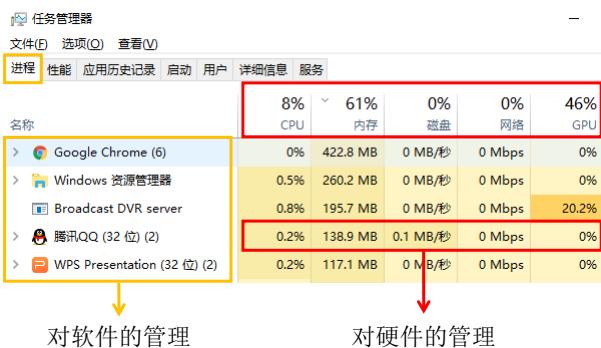
①操作系统是系统资源的管理者

操作系统 (Operating System, OS) 是指控制和管理整个计算机系统的**硬件**和**软件**资源，并合理地组织调度计算机的工作和资源的分配；以**提供给用户和其他软件方便的接口和环境**；它是计算机系统中最基本的**系统软件**。

③是最接近硬件的一层软件

②向上层提供方便易用的服务

直观的例子：打开 Windows 操作系统的“任务管理器”（快捷键：Ctrl+Alt+Del）



王道考研/CSKAOYAN.COM

操作系统的功能和目标——作为系统资源的管理者

补充知识：执行一个程序前需要将该程序放到内存中，才能被CPU处理。

用QQ和朋友视频聊天的过程：

- Step 1: 在各个文件夹中找到QQ安装的位置（如D:/Tencent/QQ/Bin）
- Step 2: 双击打开QQ.exe 需要把该程序相关数据放入内存
- Step 3: QQ程序正常运行 对应的进程被处理器（CPU）处理
- Step 4: 开始和朋友视频聊天

逐层打开文件夹，找到QQ.exe这个程序（可执行文件）的存放位置

需要将摄像头设备分配给进程

王道考研/CSKAOYAN.COM

操作系统的功能和目标——向上层提供方便易用的服务

封装思想：操作系统把一些丑陋的硬件功能封装成简单易用的服务，使用户能更方便地使用计算机，用户无需关心底层硬件的原理，只需要对操作系统发出命令即可

在硬件之上安装了操作系统，操作系统对外暴露了“美丽”、友好的交互接口

硬件只听得懂二进制指令，如：
0101010010101001010101010...

硬件对外暴露了“丑陋”、不友好的交互接口

王道考研/CSKAOYAN.COM

操作系统的功能和目标——向上层提供方便易用的服务



封装思想：操作系统把一些丑陋的硬件功能封装成简单易用的服务，使用户能更方便地使用计算机，用户无需关心底层硬件的原理，只需要对操作系统发出命令即可

王道考研/CSKAOYAN.COM

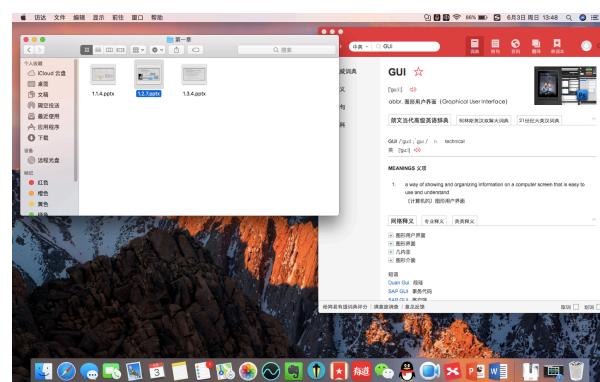
操作系统的功能和目标——向上层提供方便易用的服务

很多现代操作系统都提供GUI

GUI：图形化用户接口（Graphical User Interface）

用户可以使用形象的图形界面进行操作，而不再需要记忆复杂的命令、参数。

例子：在 Windows 操作系统中，删除一个文件只需要把文件“拖拽”到回收站即可。



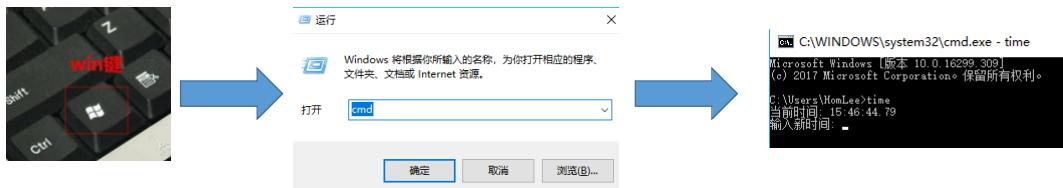
王道考研/CSKAOYAN.COM

操作系统的功能和目标——向上层提供方便易用的服务

联机命令接口实例（Windows系统） 联机命令接口=交互式命令接口

- Step 1: win键+R
- Step 2: 输入cmd, 按回车, 打开命令解释器
- Step 3: 尝试使用 time 命令

特点：用户说一句，
系统跟着做一句



王道考研/CSKAOYAN.COM

操作系统的功能和目标——向上层提供方便易用的服务

脱机命令接口实例（Windows系统） 脱机命令接口=批处理命令接口

使用windows系统的搜索功能，搜索C盘中的 *.bat文件，用记事本任意打开一个

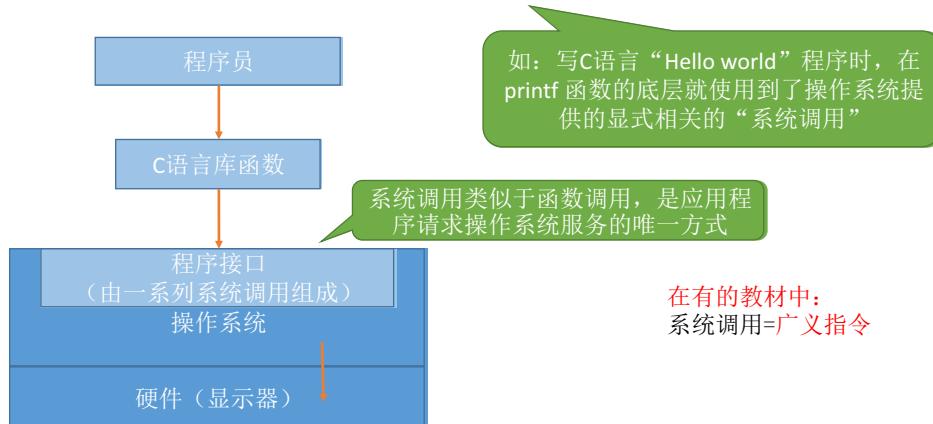
特点：用户说一堆，
系统跟着做一堆

```
1 @echo off
2
3 if '$1'=='?' goto help
4 if '$1'=='-help' goto help
5 if '$1'=='help' goto help
6 if '$1'=='-h' goto help
7
8 "%~dp0vendor\tools\NuGet.exe" install "%~dp0vendor\packages.config"
9 @SET cmd=$psakeDir = ([array](dir ""%~dp0vendor\packages\psake.*")
10 powershell -NoProfile -ExecutionPolicy Bypass -Command ^
11 goto :eof
12 |
13 :help
14 @SET cmd=$psakeDir = ([array](dir ""%~dp0vendor\packages\psake.*")
15 powershell -NoProfile -ExecutionPolicy Bypass -Command ^
16 %cmd%
```

王道考研/CSKAOYAN.COM

操作系统的功能和目标——向上层提供方便易用的服务

程序接口：可以在程序中进行**系统调用**来使用程序接口。普通用户不能直接使用程序接口，只能通过程序代码**间接使用**。



王道考研/CSKAOYAN.COM

操作系统的功能和目标——向上层提供方便易用的服务

向上层提供方便易用的服务



王道考研/CSKAOYAN.COM

操作系统的功能和目标——作为最接近硬件的层次



需要实现**对硬件机器的拓展**

没有任何软件支持的计算机成为**裸机**。在裸机上安装的操作系统，可以提供资源管理功能和方便用户的服务功能，将裸机改造成功能更强、使用更方便的机器

通常把覆盖了软件的机器成为**扩充机器**，又称之为**虚拟机**

类比汽车：

发动机——只会转；轮胎——只会滚；

在原始的硬件机器上覆盖一层传动系统——让发动机带着轮子转——使原始的硬件机器得到拓展

操作系统对硬件机器的拓展：将CPU、内存、磁盘、显示器、键盘等硬件合理地组织起来，让各种硬件能够相互协调配合，实现更多更复杂的功能

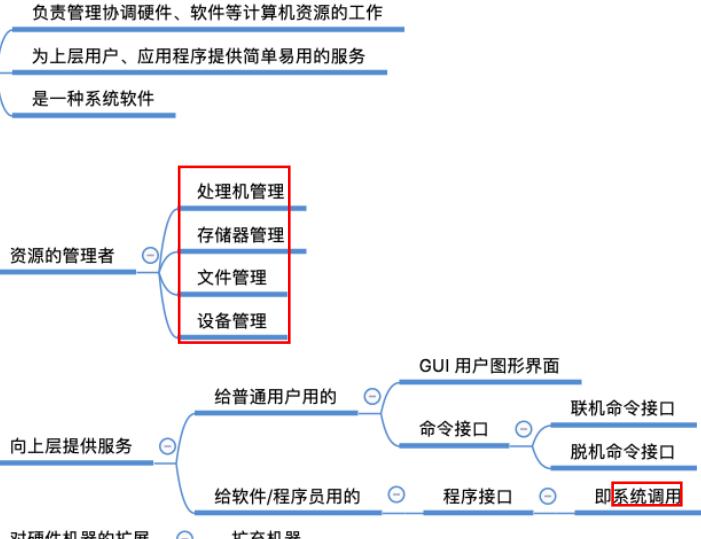
普通用户无需关心这些硬件在底层是怎么组织起来工作的，只需直接使用操作系统提供的接口即可

王道考研/CSKAOYAN.COM

知识回顾与重要考点

操作系统的...

功能和目标



王道考研/CSKAOYAN.COM

本节内容

操作系统的 四个特征

王道考研/CSKAOYAN.COM

知识总览

操作系统的特征



王道考研/CSKAOYAN.COM

操作系统的特征——并发

并发：指两个或多个事件在同一时间间隔内发生。这些事件宏观上是同时发生的，但微观上是交替发生的。
常考易混概念——**并行**：指两个或多个事件在同一时刻同时发生。

并发 VS 并行

eg：假设小渣和老渣每人有两个女朋友。任务1：和一号约会；任务2：和二号约会…



小渣



老渣



并行约会：同一时刻同时进行两个约会任务

并发约会：宏观上看，这一天老渣在同时进行两个约会任务。微观上看，在某一时刻，老渣最多正在进行一个约会任务

王道考研/CSKAOYAN.COM

操作系统的特征——并发

并发：指两个或多个事件在同一时间间隔内发生。这些事件宏观上是同时发生的，但微观上是交替发生的

操作系统的并发性指计算机系统中“同时”运行着多个程序，这些程序宏观上看是同时运行着的，而微观上看是交替运行的。

操作系统就是伴随着“多道程序技术”而出现的。因此，**操作系统和程序并发是一起诞生的**。

注意（重要考点）：

单核CPU同一时刻只能执行一个程序，各个程序只能并发地执行

多核CPU同一时刻可以同时执行多个程序，多个程序可以并行地执行

比如Intel 的第八代 i3 处理器就是**4 核CPU**，意味着可以并行地执行**4个程序**

即使是对于4核CPU来说，只要有4个以上的程序需要“同时”运行，那么并发性依然是必不可少的，因此**并发性**是操作系统一个最基本的特点



王道考研/CSKAOYAN.COM

操作系统的特征——共享

共享即资源共享，是指系统中的资源可供内存中多个并发执行的进程共同使用。

两种资源共享方式

互斥共享方式

系统中的某些资源，虽然可以提供给多个进程使用，但一个时间段内只允许一个进程访问该资源

同时共享方式

系统中的某些资源，允许一个时间段内由多个进程“同时”对它们进行访问

所谓的“同时”往往是宏观上的，而在微观上，这些进程可能是交替地对该资源进行访问的（即分时共享）

生活实例：

互斥共享方式：使用QQ和微信视频。同一时间段内摄像头只能分配给其中一个进程。

同时共享方式：使用QQ发送文件A，同时使用微信发送文件B。宏观上看，两边都在同时读取并发送文件，说明两个进程都在访问硬盘资源，从中读取数据。微观上看，两个进程是交替着访问硬盘的。

王道考研/CSKAOYAN.COM

操作系统的特征——并发和共享的关系

并发性指计算机系统中同时存在着多个运行着的程序。

共享性是指系统中的资源可供内存中多个并发执行的进程共同使用。

通过上述例子来看并发与共享的关系：

使用QQ发送文件A，同时使用微信发送文件B。

如果失去并发性，则系统中只有一个程序正在运行，则共享性失去存在的意义

1. 两个进程正在并发执行（并发性）

如果失去共享性，则QQ和微信不能同时访问硬盘资源，就无法实现同时发送文件，也就无法并发

2. 需要共享地访问硬盘资源（共享性）

伸个懒腰都是爱你的形状

并发性



共享性

王道考研/CSKAOYAN.COM

操作系统的特征——虚拟

虚拟是指把一个物理上的实体变为若干个逻辑上的对应物。物理实体（前者）是实际存在的，而逻辑上对应物（后者）是用户感受到的。

Yo~用一个例子来理解

背景知识：一个程序需要放入内存并给它分配CPU才能执行



GTAV需要4GB的运行内存，QQ需要256MB的内存，迅雷需要256MB的内存.....

我的电脑：4GB内存

问题：这些程序同时运行需要的内存远大于4GB，那么为什么它们还可以在我的电脑上同时运行呢？

答：这是虚拟存储器技术。实际只有4GB的内存，在用户看来似乎远远大于4GB



王道考研/CSKAOYAN.COM

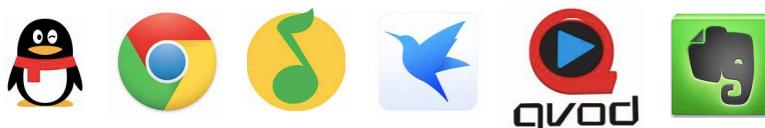
操作系统的特征——虚拟

虚拟是指把一个物理上的实体变为若干个逻辑上的对应物。物理实体（前者）是实际存在的，而逻辑上对应物（后者）是用户感受到的。

Yo~用一个例子来理解

背景知识：一个程序需要放入内存并给它分配CPU才能执行

某单核CPU的计算机中，用户打开了以下软件。。。



王道考研/CSKAOYAN.COM

问题：既然一个程序需要被分配CPU才能正常执行，那么为什么单核CPU的电脑中能同时运行这么多程序呢？

答：这是虚拟处理器技术。实际上只有一个单核CPU，在用户看来似乎有6个CPU在同时为自己服务

操作系统的特征——虚拟

虚拟是指把一个物理上的实体变为若干个逻辑上的对应物。物理实体（前者）是实际存在的，而逻辑上对应物（后者）是用户感受到的。

虚拟技术

空分复用技术（如虚拟存储器技术）

时分复用技术（如虚拟处理器）

显然，如果失去了并发性，则一个时间段内系统中只需运行一道程序，那么就失去了实现虚拟性的意义了。因此，**没有并发性，就谈不上虚拟性**

王道考研/CSKAOYAN.COM

操作系统的特征——异步

异步是指，在多道程序环境下，允许多个程序并发执行，但由于资源有限，进程的执行不是一贯到底的，而是走走停停，以不可预知的速度向前推进，这就是进程的异步性。

老渣要和两个女孩并发约会



一号的指令1: 老渣陪我吃饭
一号的指令2: 老渣把心给我] 第一道程序

二号的指令1: 老渣把心给我
二号的指令2: 老渣陪我吃饭] 第二道程序

与一、二号的约会 = 两道并发执行的程序
老渣的心 = 有限的系统资源

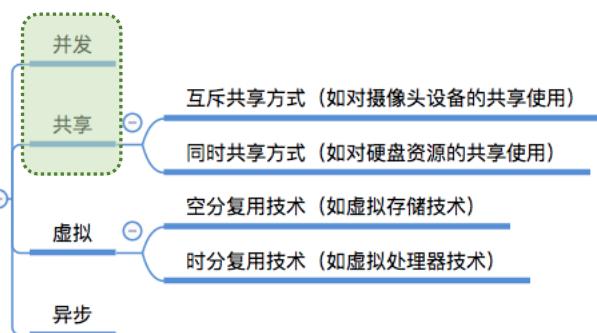
由于并发运行的程序会争抢着使用系统资源，而系统中的资源有限，因此进程的执行不是一贯到底的，而是走走停停的，以不可预知的速度向前推进

如果失去了并发性，即系统只能串行地运行各个程序，那么每个程序的执行会一贯到底。**只有系统拥有并发性，才有可能导致异步性。**

王道考研/CSKAOYAN.COM

知识回顾与重要考点

操作系统的特征



重要考点：

理解并发和并行的区别

并发和共享互为存在条件

没有并发和共享，就谈不上虚拟和异步，因此并发和共享是操作系统的两个最基本的特征

王道考研/CSKAOYAN.COM

本节内容

操作系统的 发展与分类

王道考研/CSKAOYAN.COM

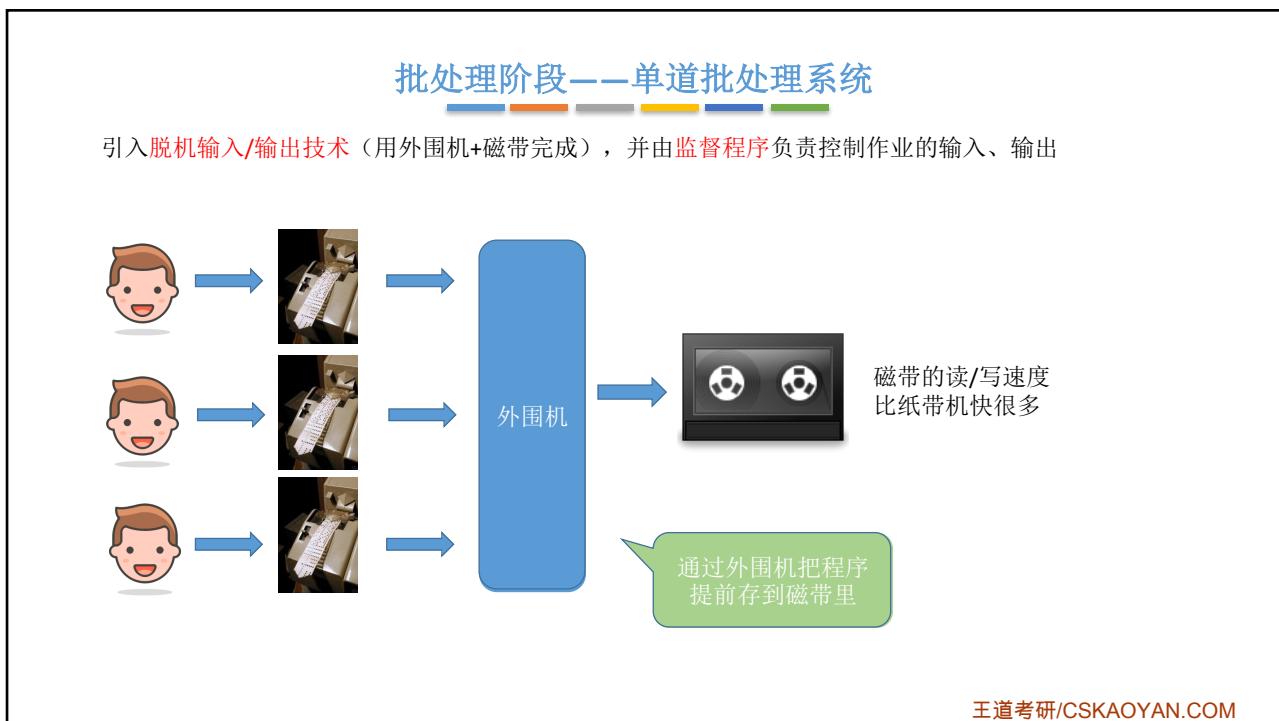
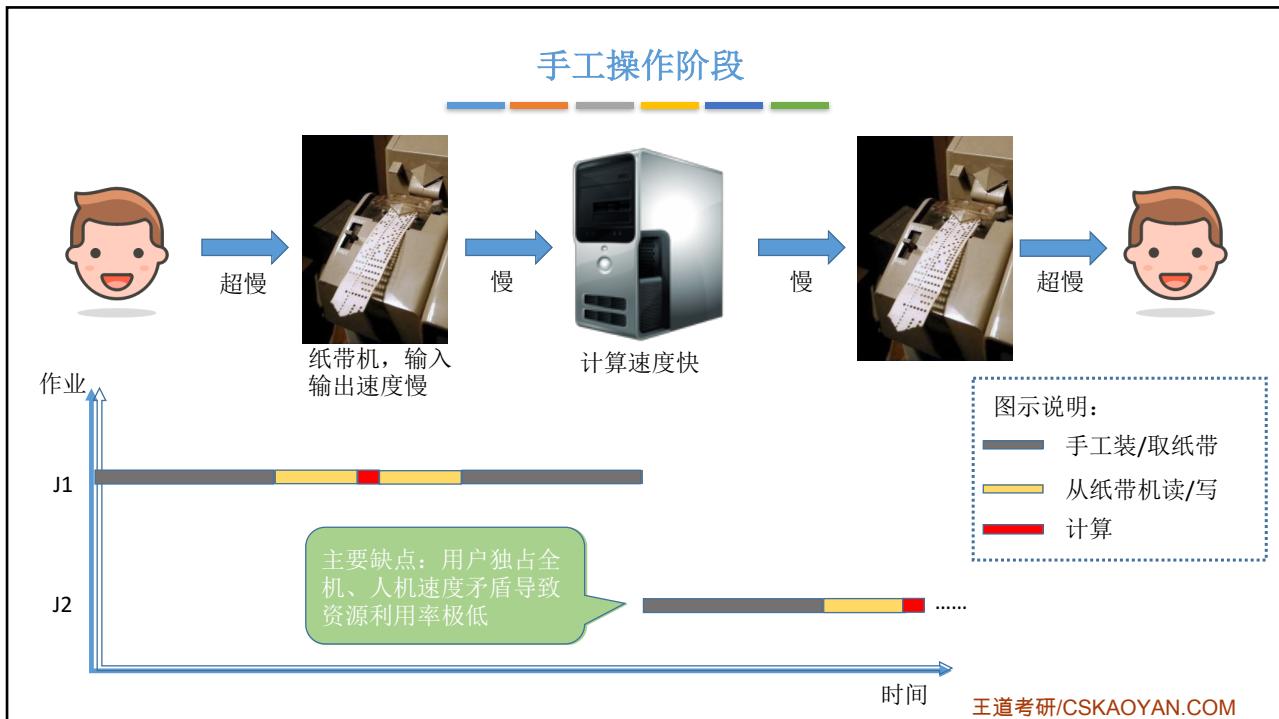
知识总览

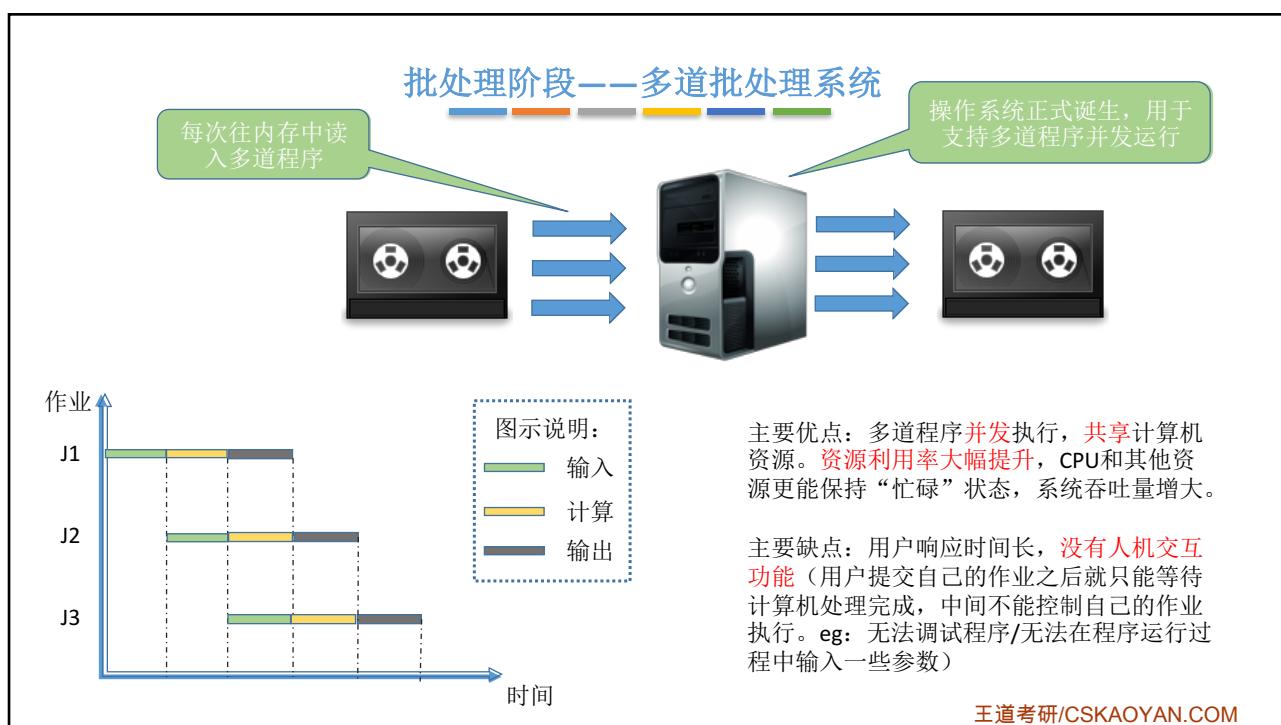
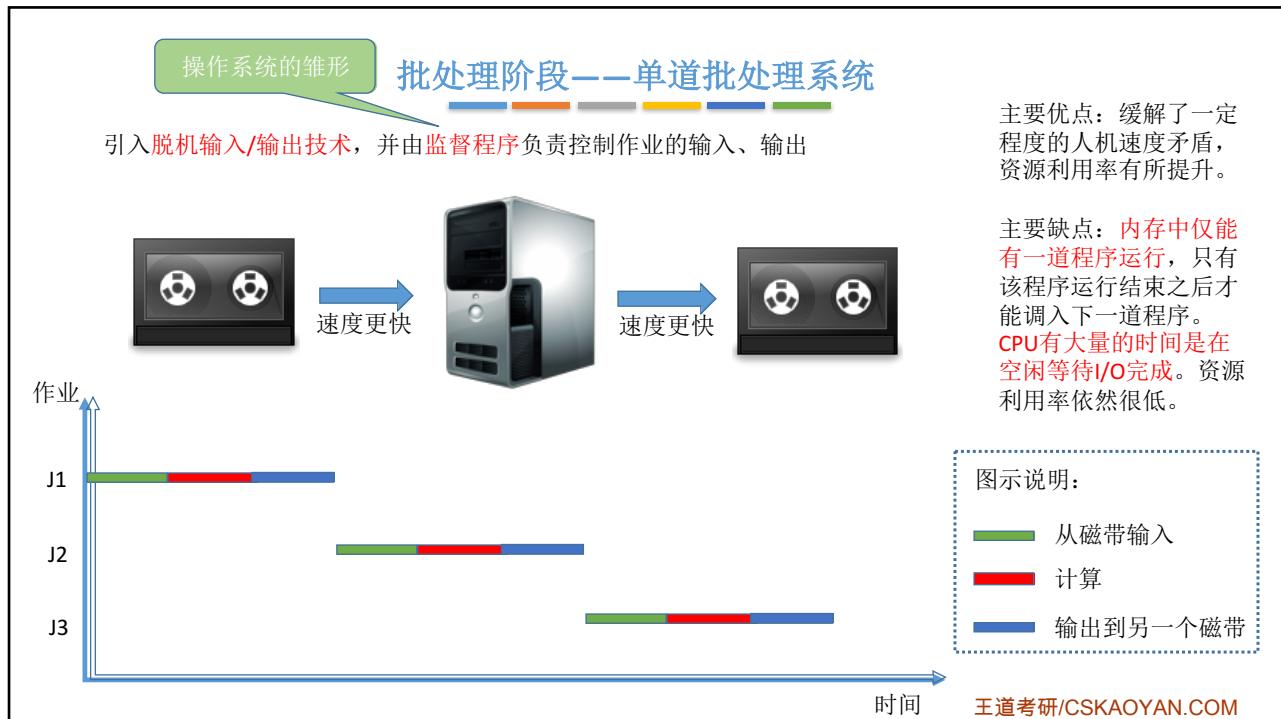
OS的发展与分类



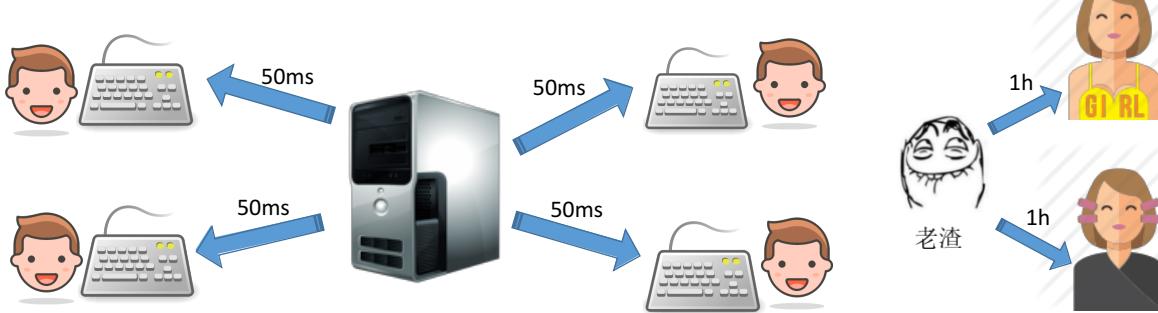
学习提示：要重点关注和理解各类操作系统主要想解决的是什么问题，各自的优缺点。

王道考研/CSKAOYAN.COM





分时操作系统



分时操作系统：计算机以**时间片**为单位**轮流为各个用户/作业服务**，各个用户可通过终端与计算机进行交互。

主要优点：用户请求可以被即时响应，**解决了人机交互问题**。允许多个用户同时使用一台计算机，并且用户对计算机的操作相互独立，感受不到别人的存在。

主要缺点：**不能优先处理一些紧急任务**。操作系统对各个用户/作业都是完全公平的，循环地为每个用户/作业服务一个时间片，不区分任务的紧急性。

王道考研/CSKAOYAN.COM

实时操作系统

实时操作系统：

主要优点：能够优先响应一些紧急任务，某些紧急任务不需时间片排队。

在实时操作系统的控制下，计算机系统接收到外部信号后及时进行处理，并且**要在严格的时限内处理完事件**。实时操作系统的主要特点是**及时性和可靠性**

如：导弹控制系统、
自动驾驶系统

实时操作系统

硬实时系统 ⊕ 必须在绝对严格的规定时间内完成处理

软实时系统 ⊕ 能接受偶尔违反时间规定

如：12306火车订票系统

王道考研/CSKAOYAN.COM

其他几种操作系统

网络操作系统：是伴随着计算机网络的发展而诞生的，能把网络中各个计算机有机地结合起来，实现数据传送等功能，**实现网络中各种资源的共享（如文件共享）和各台计算机之间的通信**。（如：Windows NT 就是一种典型的网络操作系统，网站服务器就可以使用）

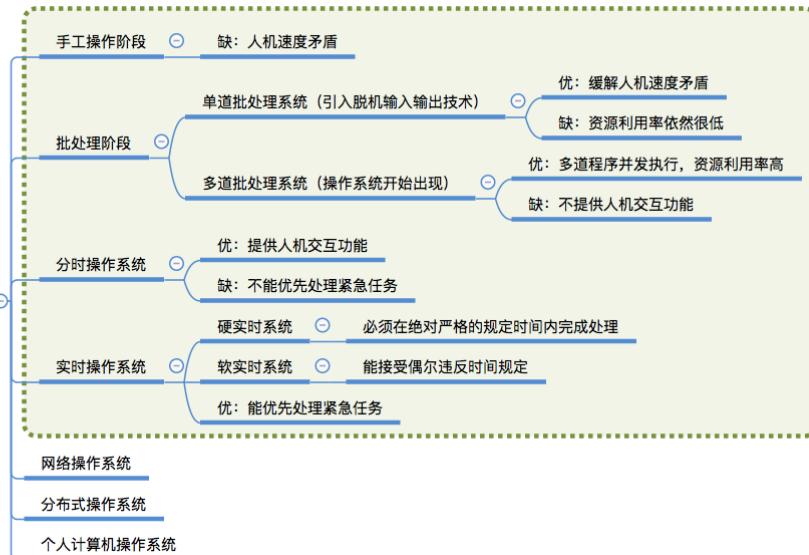
分布式操作系统：主要特点是**分布性和并行性**。系统中的各台计算机地位相同，**任何工作都可以分布在这些计算机上，由它们并行、协同完成这个任务**。

个人计算机操作系统：如 Windows XP、MacOS，方便个人使用。

王道考研/CSKAOYAN.COM

知识回顾与重要考点

OS的发展与分类



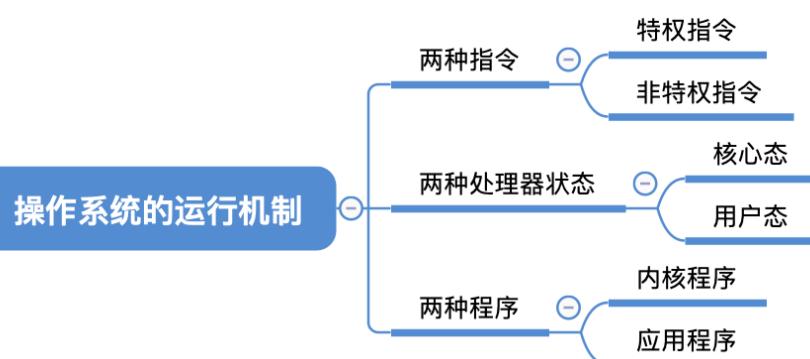
王道考研/CSKAOYAN.COM

本节内容

操作系统的运行机制

王道考研/CSKAOYAN.COM

知识总览



王道考研/CSKAOYAN.COM

预备知识：程序是如何运行的？

The diagram illustrates the compilation process. On the left, a green circular icon with a brain and question marks is shown. Next to it, a blue scroll labeled "C语言代码" (C language code) is being processed by a green arrow labeled "编译器" (Compiler) and "翻译" (Translation) into a blue box labeled "机器指令 (二进制)" (Machine Instructions (Binary)). A green speech bubble to the right states: "一条高级语言的代码翻译过来可能会对应多条机器指令" (One line of high-level language code may be translated into multiple machine instructions).

示例代码：

```
Int x = 1;
x++;
```

该代码被翻译为以下机器指令：

```
100010101100001100011100
001011000000001101001111
100100100000001100000001
001011000100111100000011
```

注释：

- “小黑框”中使用的命令也是最基本命令。
- 注意与本节的“指令”区别。

王道考研/CSKAOYAN.COM

预备知识：程序是如何运行的？

The diagram illustrates the compilation process. On the left, a green circular icon with a brain and question marks is shown. Next to it, a blue scroll labeled "C语言代码" (C language code) is being processed by a green arrow labeled "编译器" (Compiler) and "翻译" (Translation) into a blue box labeled "机器指令 (二进制)" (Machine Instructions (Binary)). A green speech bubble to the right states: "一条高级语言的代码翻译过来可能会对应多条机器指令" (One line of high-level language code may be translated into multiple machine instructions).

示例代码：

```
Int x = 1;
x++;
```

该代码被翻译为以下机器指令：

```
100010101100001100011100
001011000000001101001111
100100100000001100000001
001011000100111100000011
```

注释：

- “指令”就是处理器（CPU）能识别、执行的最基本命令。
- 注：很多人习惯把 Linux、Windows、MacOS 的“小黑框”中使用的命令也称为“指令”，其实这是“交互式命令接口”，注意与本节的“指令”区别开。本节中的“指令”指二进制机器指令。

王道考研/CSKAOYAN.COM

内核程序 v.s. 应用程序

The diagram illustrates the compilation process. On the left, there's a green circular icon with a question mark and a pencil. Next to it is a blue scroll labeled "C语言代码". A green arrow labeled "编译器" and "翻译" (translation) points to a blue box labeled "机器指令 (二进制)". To the right of this box is a green speech bubble containing the text: "一条高级语言的代码翻译过来可能会对应多条机器指令". Below this, another blue scroll shows the C code "Int x = 1; x++;" which points to a blue box containing binary machine code: 100010101100001100011100, 001011000000001101001111, 100100100000001100000001, 001011000100111100000011.

我们普通程序员写的程序就是“**应用程序**”

微软、苹果有一帮人负责实现操作系统，他们写的是“**内核程序**”
由很多内核程序组成了“**操作系统内核**”，或简称“**内核（Kernel）**”
内核是操作系统最重要最核心的部分，也是**最接近硬件的部分**
甚至可以说，一个操作系统只要有内核就够了（eg: Docker→仅需Linux内核）
操作系统的功能未必都在内核中，如图形化用户界面 GUI

王道考研/CSKAOYAN.COM

特权指令 v.s. 非特权指令

The diagram illustrates the compilation process. On the left, there's a blue scroll labeled "Int x = 1; x++;" which points to a blue box containing binary machine code: 100010101100001100011100, 001011000000001101001111, 100100100000001100000001, 001011000100111100000011.

应用程序只能使用“**非特权指令**”，如：
加法指令、减法指令等

我们普通程序员写的程序就是“**应用程序**”

微软、苹果有一帮人负责实现操作系统，他们写的就是“**内核程序**”

操作系统内核作为“管理者”，有时会让CPU执行一些“**特权指令**”，如：内存清零指令。这些指令影响重大，只允许“管理者”——即操作系统内核来使用

程序运行的过程其实都是CPU执行一条一条的机器指令的过程

在**CPU设计和生产的时候**就划分了**特权指令**和**非特权指令**，因此**CPU**执行一条指令前就能判断出其类型

王道考研/CSKAOYAN.COM

内核态 v.s. 用户态

A diagram illustrating the execution process. On the left, a blue box contains C++ code: `Int x = 1;
x++;`. An arrow points from this code to a larger blue box containing binary machine code: `100010101100001100011100
001011000000001101001111
100100100000001100000001
001011000100111100000011`. To the right is a green CPU icon. Below the code box is a cartoon panda face with a speech bubble asking: "CPU 能判断出指令类型, 但是它怎么区分此时正在运行的是内核程序 or 应用程序?". A green box to the right states: "程序运行的过程其实就是CPU执行一条一条的机器指令的过程". Another orange speech bubble asks: "问题: 如何实现CPU状态的切换?". Text at the bottom explains: "CPU 有两种状态, “内核态” 和 “用户态”
处于内核态时, 说明此时正在运行的是内核程序, 此时可以执行特权指令
处于用户态时, 说明此时正在运行的是应用程序, 此时只能执行非特权指令".

拓展: CPU 中有一个寄存器叫 **程序状态字寄存器 (PSW)**, 其中有个二进制位, 1 表示“内核态”, 0 表示“用户态”
别名: 内核态=核心态=管态; 用户态=目态

王道考研/CSKAOYAN.COM

A diagram showing the state transition of the CPU. It starts with two boxes: "内核程序" (Orange) and "应用程序" (Blue). Arrows point from both to a central column of binary code. The top row of code is orange, and the bottom row is blue. To the right, there's a green CPU icon, a blue starburst labeled "用户态", an orange starburst labeled "内核态", and a red starburst labeled "中断信号". A cartoon character with a sword says "拒绝!". A green box contains text: "操作系统内核在让出 CPU 之前, 会用一条特权指令把 PSW 的标志位设置为“用户态”". A green arrow points from the text to the "User Mode" starburst. A green box at the bottom right says: "CPU 检测到中断信号后, 会立即变为“核心态”, 并停止运行当前的应用程序, 转而运行处理中断信号的内核程序". A list of 9 numbered points describes the process:

- ① 处理中断信号的内核程序为“内核态”;
- ② 应用程序, 用户可以启动某个...
- ③ 操作系统内核程序在合适的时候主动让出 CPU, 让该应用程序手工 CPU 运行
- ④ 应用程序运行在“用户态”
- ⑤ 此时, 一位猥琐黑客在应用程序中植入了一条特权指令, 企图破坏系统...
- ⑥ CPU 发现接下来要执行的这条指令是特权指令, 但是自己又处于“用户态”
- ⑦ 这个非法事件会引发一个中断信号
- ⑧ “中断”使操作系统再次夺回 CPU 的控制权
- ⑨ 操作系统会对引发中断的事件进行处理, 处理完了再把 CPU 使用权交给别的应用程序

王道考研/CSKAOYAN.COM

内核态、用户态的切换

内核态→用户态：执行一条特权指令——修改PSW的标志位为“用户态”，这个动作意味着操作系统将主动让出CPU使用权

用户态→内核态：由“中断”引发，硬件自动完成变态过程，触发中断信号意味着操作系统将强行收回CPU的使用权

除了非法使用特权指令之外，还有很多事件会触发中断信号。一个共性是，**但凡需要操作系统介入的地方，都会触发中断信号**

一个故事：

- ① 刚开机时，CPU为“**内核态**”，操作系统内核程序先上CPU运行
- ② 开机完成后，用户可以启动某个应用程序
- ③ 操作系统内核程序在合适的时候主动让出CPU，让该应用程序上CPU运行
- ④ 应用程序运行在“**用户态**”
- ⑤ 此时，一位猥琐黑客在应用程序中植入了一条特权指令，企图破坏系统...
- ⑥ CPU发现接下来要执行的这条指令是特权指令，但是自己又处于“**用户态**”
- ⑦ 这个非法事件会引发一个**中断信号**

操作系统内核在让出CPU之前，会用一条特权指令把PSW的标志位设置为“**用户态**”

CPU检测到中断信号后，会立即变为“**核心态**”，并停止运行当前的应用程序，转而运行处理中断信号的内核程序

- ⑧ “**中断**”使操作系统再次收回CPU的控制权
- ⑨ 操作系统会对引发中断的事件进行处理，处理完了再把CPU使用权交给别的应用程序

王道考研/CSKAOYAN.COM

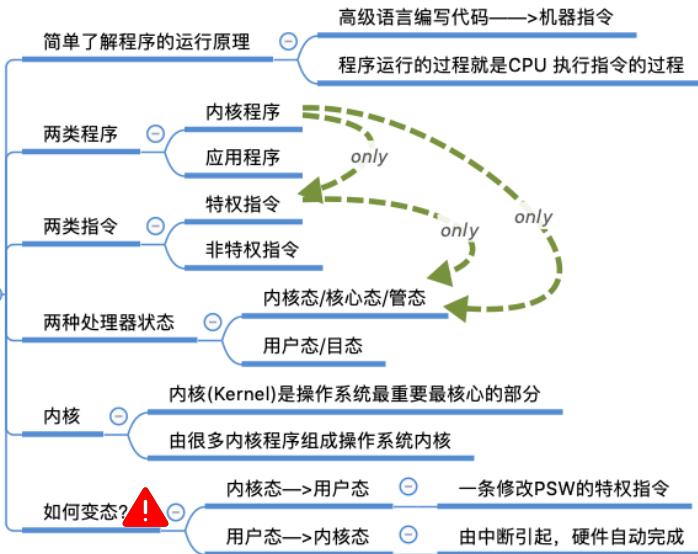
知识回顾与重要考点



操作系统的运行机制

Tips:

1. 都是高频考点，很重要
2. 初学者不完全理解没关系，放心大胆地往后学，随着后面章节的学习，理解会逐渐加深



王道考研/CSKAOYAN.COM

两种指令、两种处理器状态、两种程序



新的问题：
有的指令“人畜无害”。比如：加、减、乘、除这些普通的运算指令。
有的指令有很高的权限。比如：内存清零指令。如果用户程序可以使用这个指令，就意味着一个用户可以将其他用户的内存数据随意清零，这样做显然是很危险的。

指令

- 特权指令：如内存清零指令
不允许用户程序使用
- 非特权指令：如普通的运算指令

王道考研/CSKAOYAN.COM

两种指令、两种处理器状态、两种程序



问题：CPU如何判断当前是否可以执行特权指令？

两种处理器状态

- 用户态（目态）
此时CPU只能执行非特权指令
- 核心态（管态）
特权指令、非特权指令都可执行

用程序状态字寄存器（PSW）中的某标志位来标识当前处理器处于什么状态。如0为用户态，1为核心态

王道考研/CSKAOYAN.COM

两种指令、两种处理器状态、两种程序

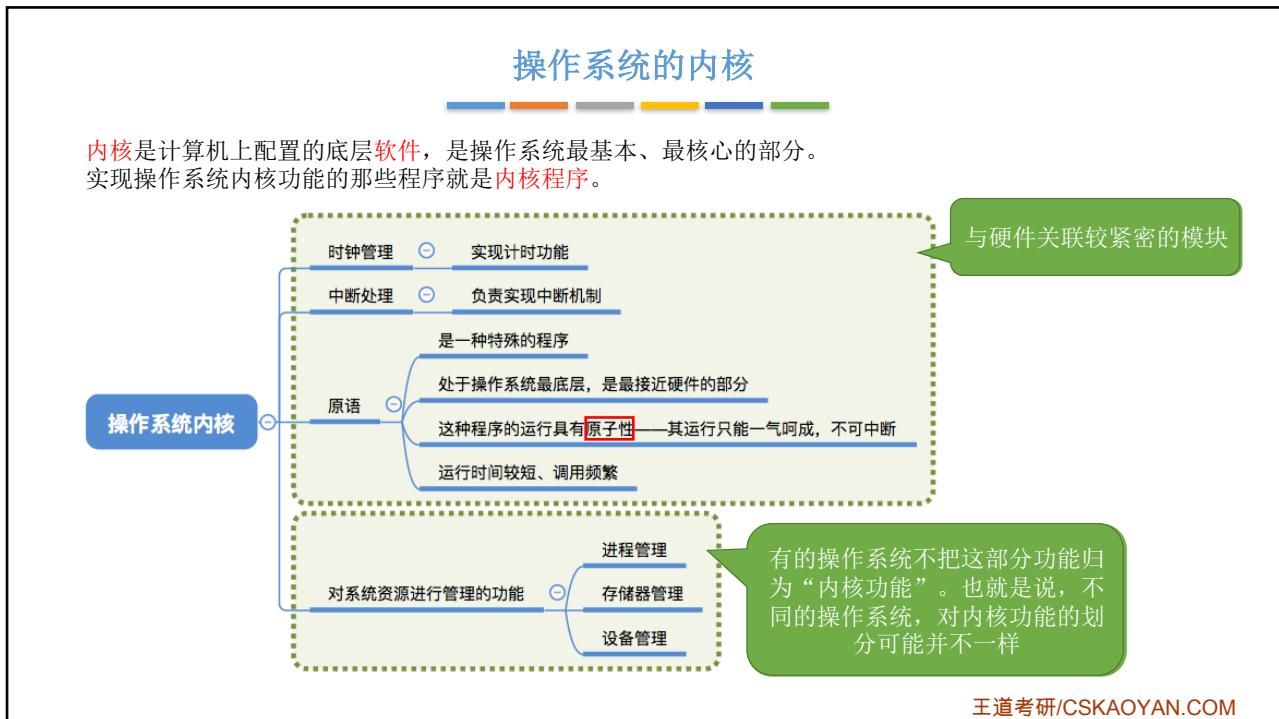
The diagram illustrates the relationship between two types of programs: **内核程序** (Kernel Programs) and **应用程序** (User Applications). A blue rounded rectangle labeled **两种程序** (Two Types of Programs) contains two smaller boxes: one for **内核程序** and one for **应用程序**. A speech bubble next to the **内核程序** box states: "操作系统的内核程序是系统的管理者，既可以执行特权指令，也可以执行非特权指令，运行在核心态。" (The operating system's kernel program is the system manager, which can execute both privileged and unprivileged instructions, running in the core state.) Another speech bubble next to the **应用程序** box states: "为了保证系统能安全运行，普通应用程序只能执行非特权指令，运行在用户态" (To ensure system safety, ordinary applications can only execute non-privileged instructions, running in the user state).

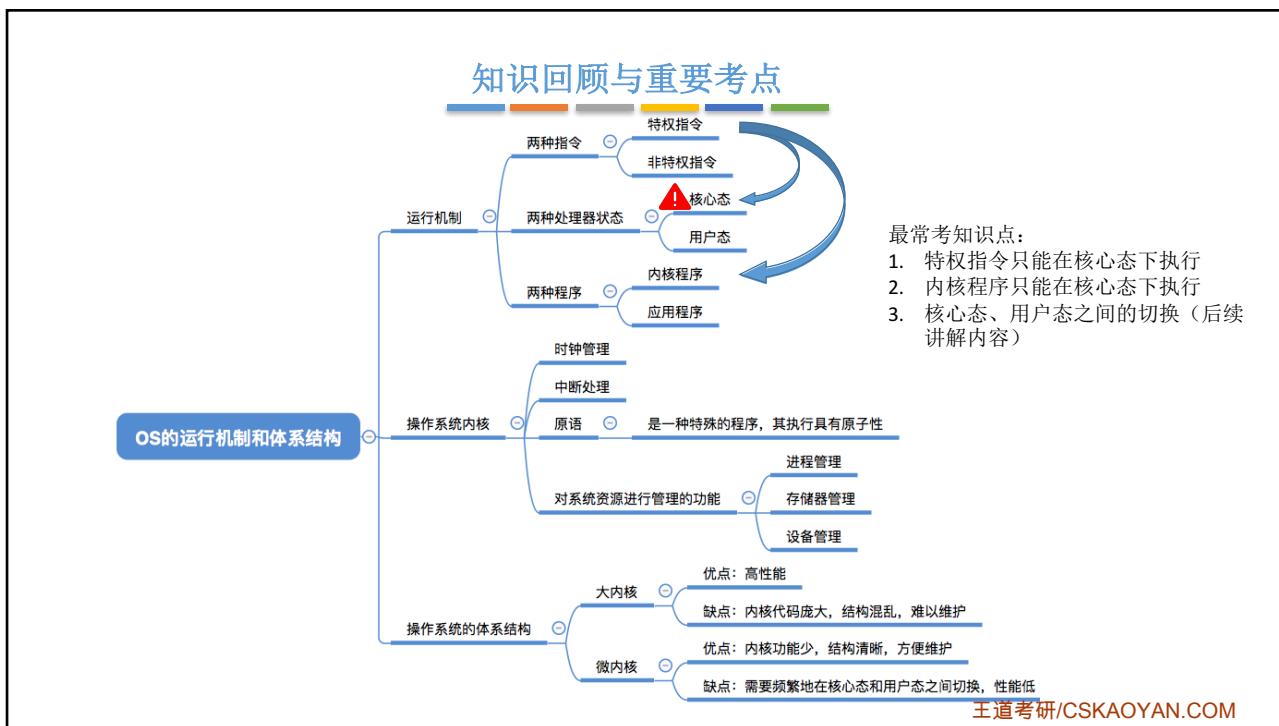
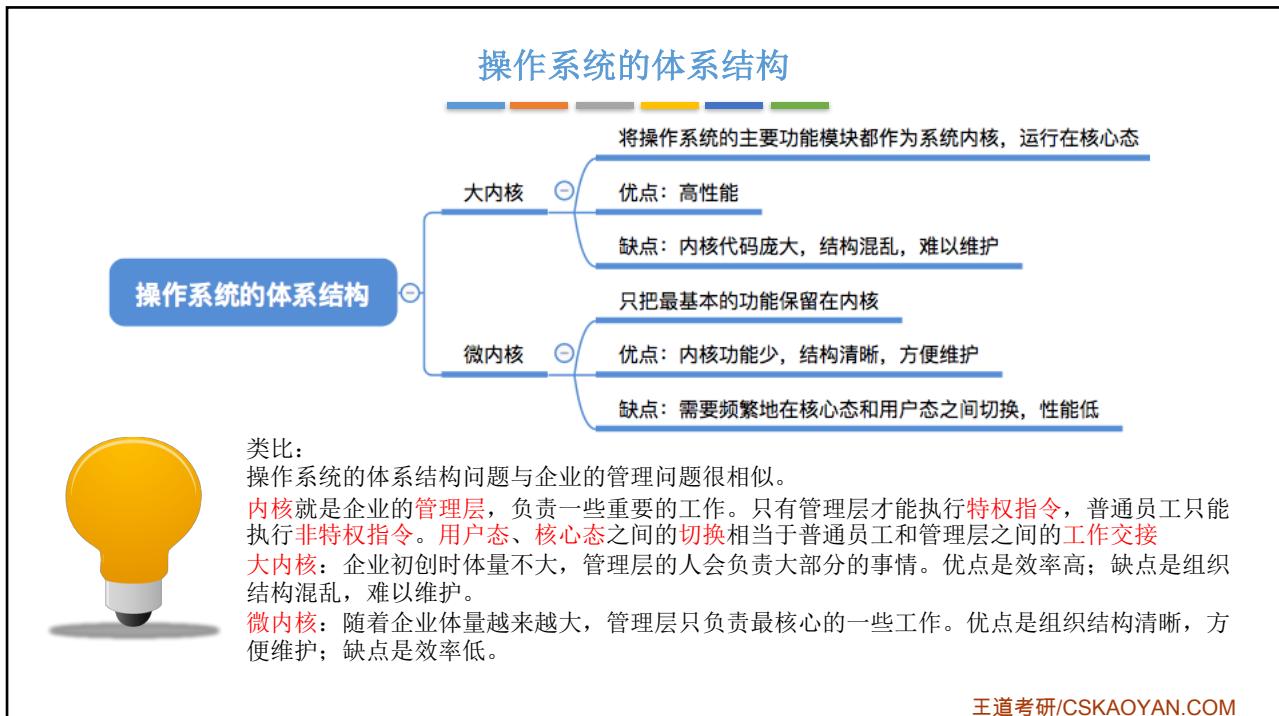
王道考研/CSKAOYAN.COM

操作系统的内核

The diagram shows the **计算机系统的层次结构** (Hierarchical Structure of Computer Systems). It is divided into two main layers: **操作系统** (Operating System) and **裸机 (纯硬件)** (Naked Machine (Pure Hardware)). The **操作系统** layer is further divided into **用户**, **应用程序 (软件)**, **非内核功能**, and **进程管理、存储器管理、设备管理等功能**. The **裸机** layer includes **时钟管理**, **中断处理**, and **原语 (设备驱动、CPU切换等)**. A bracket on the left indicates that the **实现计时功能** (Achieve timing function) is part of the operating system. A bracket on the right indicates that the **原语** (Device driver, CPU switching, etc.) is part of the kernel. A speech bubble explains: "原语是一种特殊的程序。是最接近硬件的部分，这种程序的运行具有原子性。" (A macroinstruction is a special type of program. It is the part closest to hardware, and its execution has atomicity.)

王道考研/CSKAOYAN.COM





本节内容

中断和异常

王道考研/CSKAOYAN.COM

知识总览



中断的作用

中断的类型

内中断（也称“异常”）

外中断

中断机制的基本原理

中断和异常

王道考研/CSKAOYAN.COM

中断的作用

CPU 上会运行两种程序，一种是**操作系统内核程序**，一种是**应用程序**

是整个系统的
管理者

“中断”会使CPU由用
户态变为内核态，使操
作系统重新夺回对CPU
的控制权

在合适的情况下，操作系统内核会把CPU的使用权主动让给应用程序（第二章进程管理相关内容）

“中断”是**让操作系统内核夺回CPU使用权**的唯一途径

如果没有“中断”机制，那么一旦应用程序上CPU运行，CPU就会一直运行这个应用程序



内核态→用户态：执行一条**特权指令**——修改PSW的标志位为“用户态”，这个动作意味着操作系统将主动让出CPU使用权

用户态→内核态：由“中断”引发，硬件自动完成变态过程，触发中断信号意味着操作系统将强行夺回CPU的使用权

王道考研/CSKAOYAN.COM

中断的类型

中断的类型

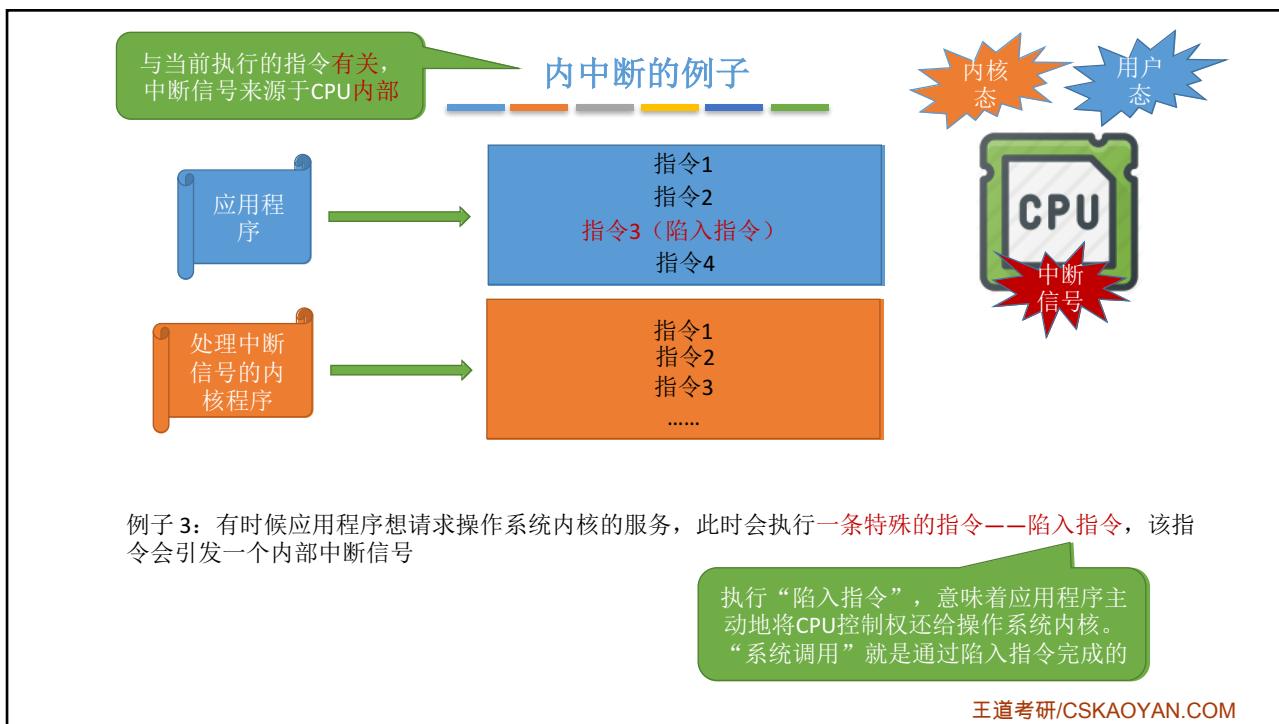
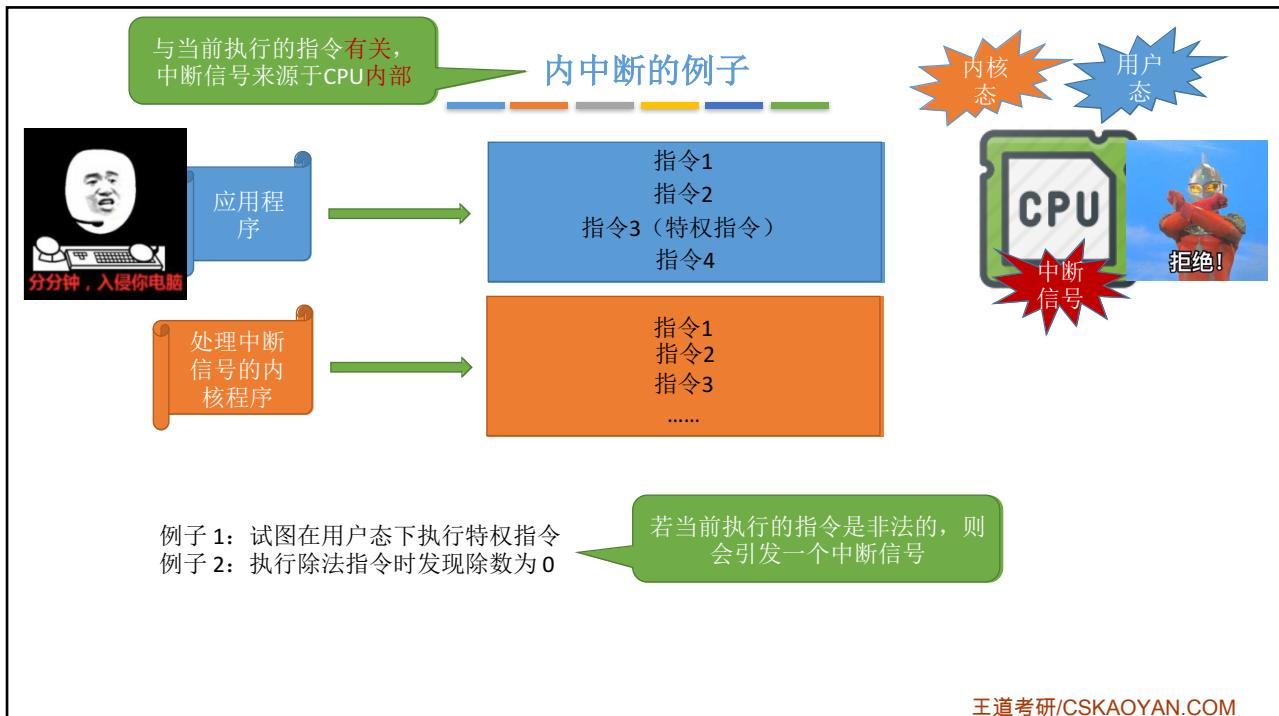
内中断

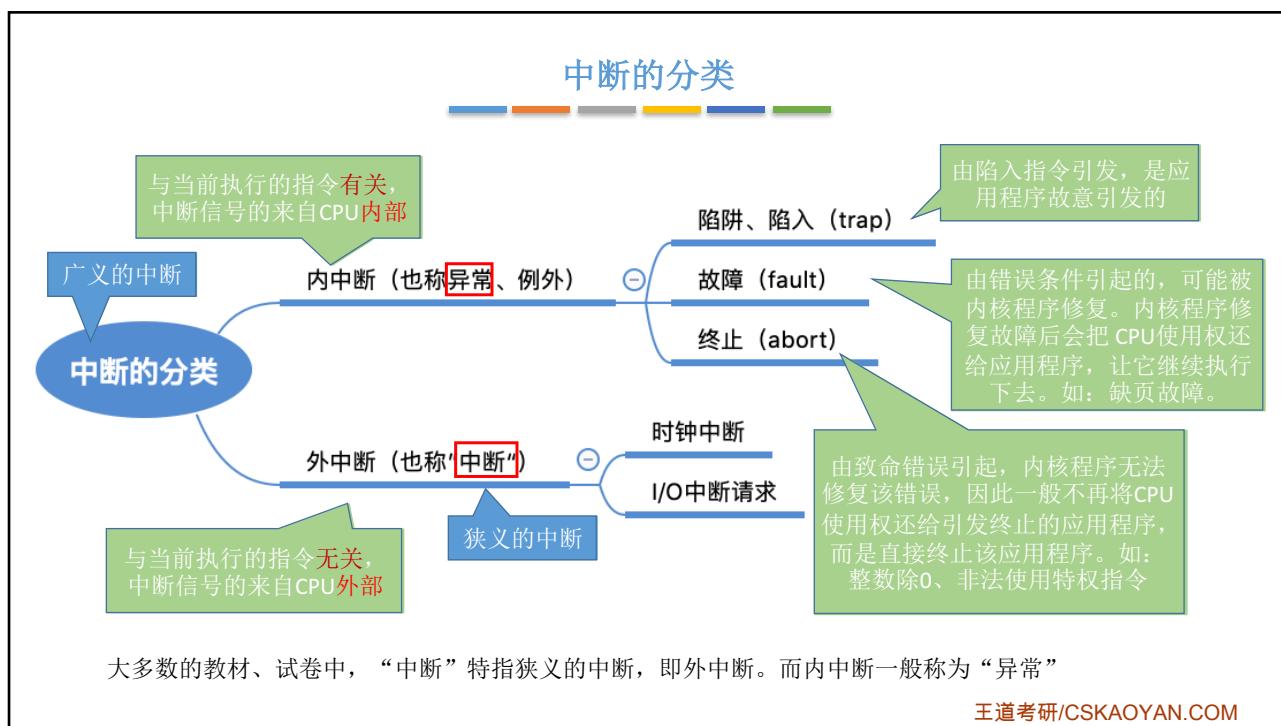
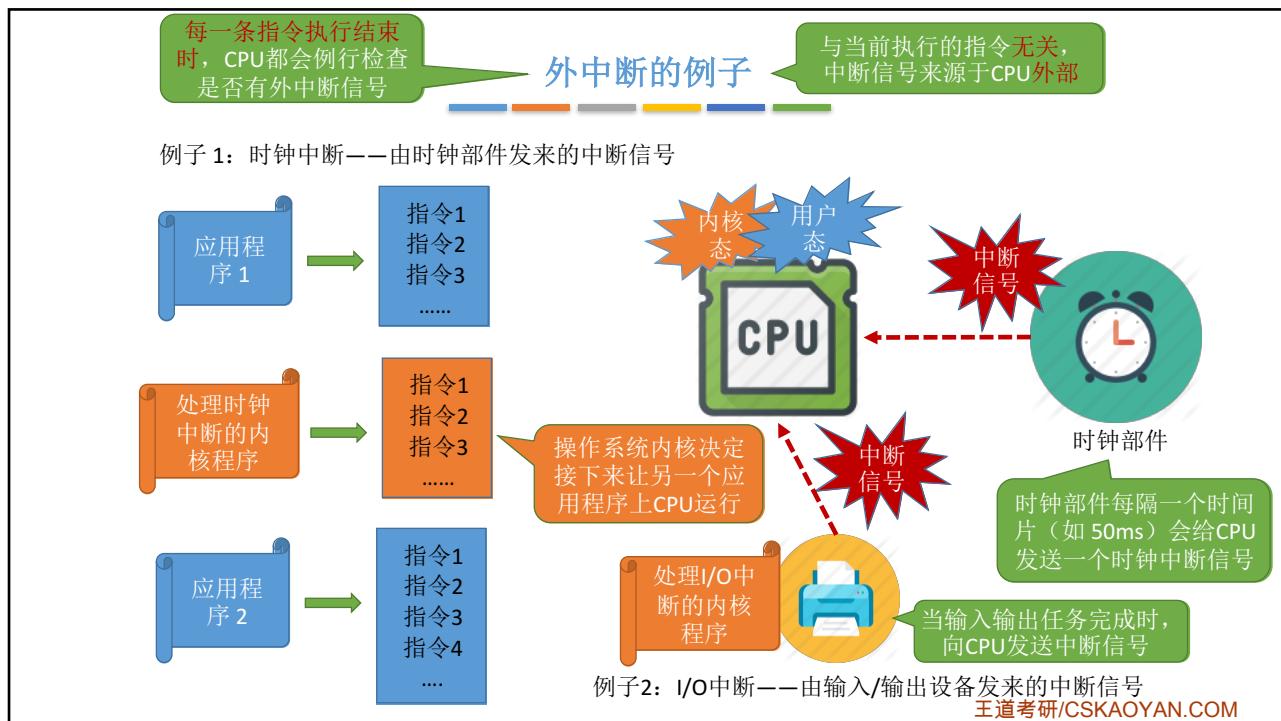
与当前执行的指令**有关**，
中断信号来源于CPU**内部**

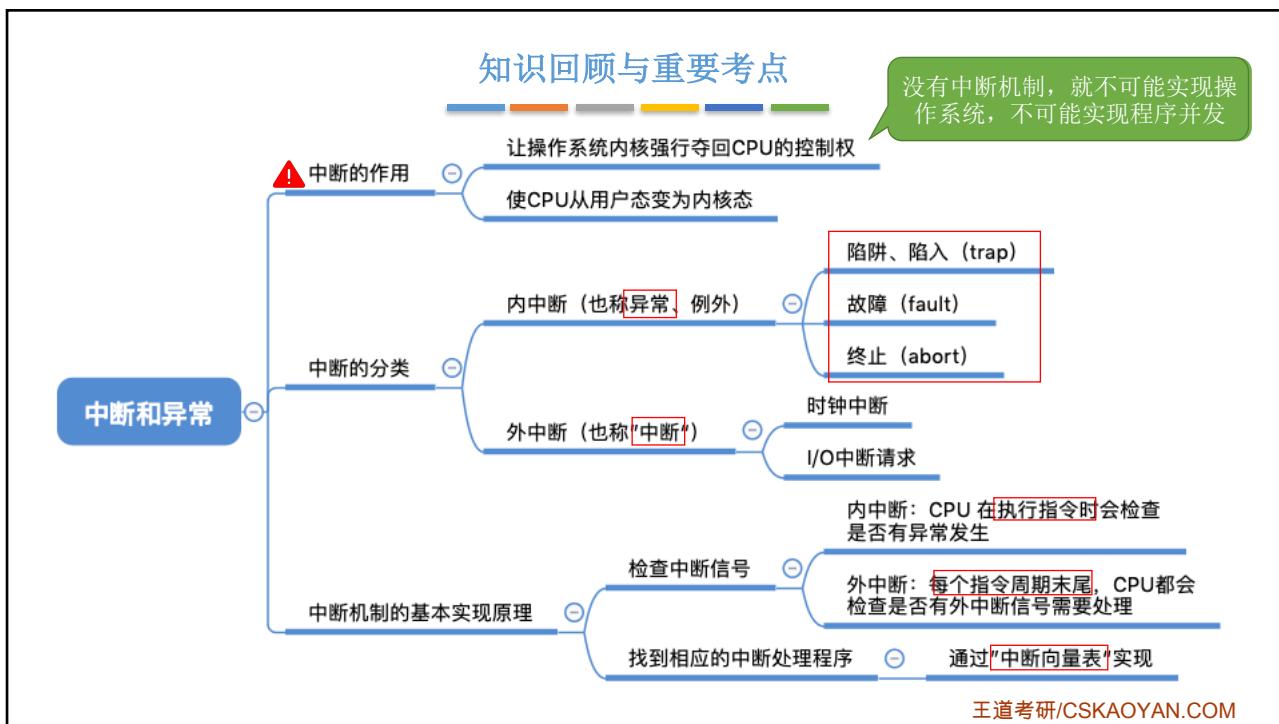
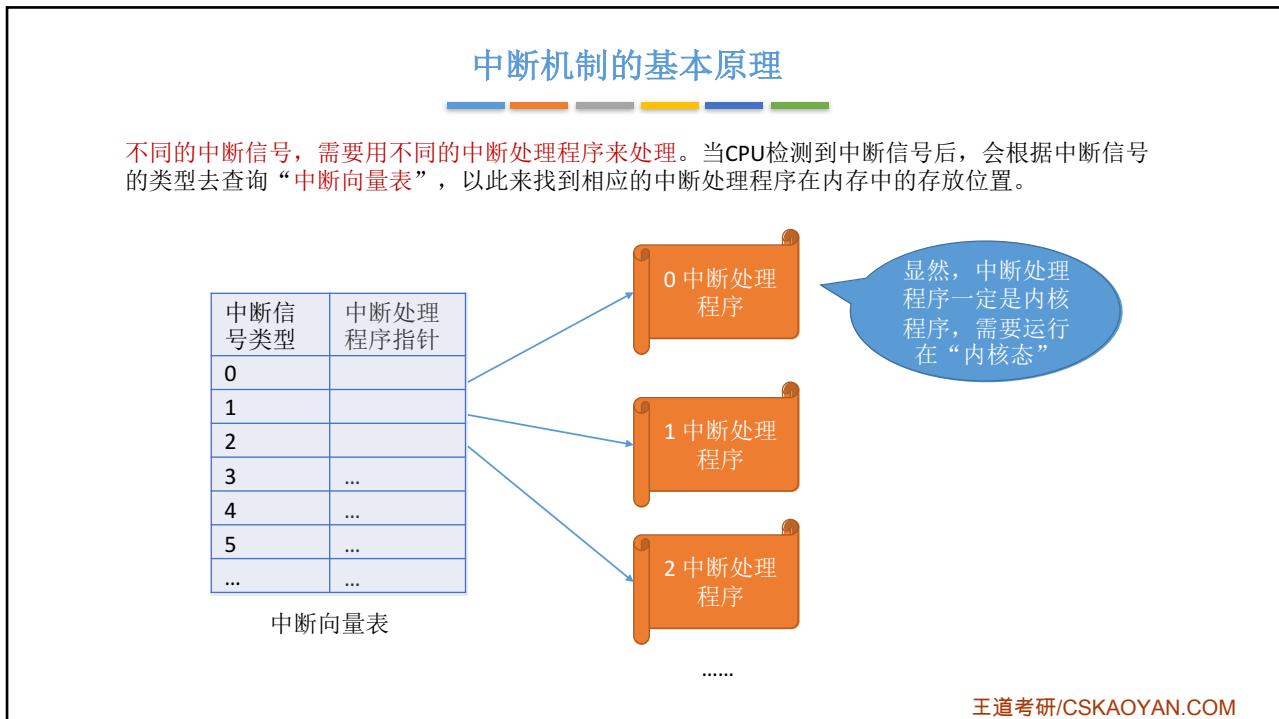
外中断

与当前执行的指令**无关**，
中断信号来源于CPU**外部**

王道考研/CSKAOYAN.COM







本节内容

系统调用

王道考研/CSKAOYAN.COM

知识总览

- 
- 什么是系统调用？
 - 系统调用与库函数的区别
 - 小例子：为什么系统调用是必须的？
 - 什么功能要用系统调用实现？
 - 系统调用的过程

王道考研/CSKAOYAN.COM

什么是系统调用，有何作用？

知识点回顾：

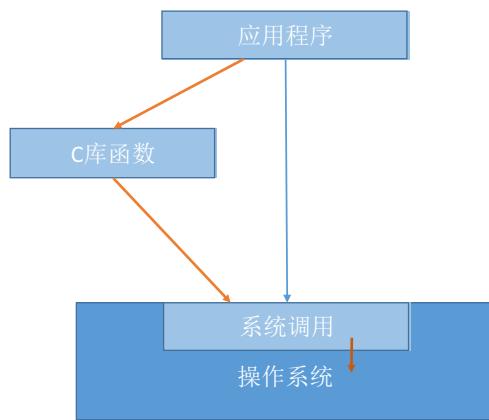
操作系统作为用户和计算机硬件之间的接口，需要向上提供一些简单易用的服务。主要包括命令接口和程序接口。其中，程序接口由一组**系统调用**组成。



“系统调用”是操作系统提供给应用程序（程序员/编程人员）使用的接口，可以理解为一种可供应用程序调用的特殊函数，**应用程序可以通过系统调用来请求获得操作系统内核的服务**

王道考研/CSKAOYAN.COM

系统调用与库函数的区别



普通应用程序	可直接进行系统调用，也可使用库函数。有的库函数涉及系统调用，有的不涉及
编程语言	向上提供库函数。有时会将系统调用封装成库函数，以隐藏系统调用的一些细节，使程序员编程更加方便。
操作系统	向上提供系统调用，使得上层程序能请求内核的服务
裸机	

不涉及系统调用的库函数：如的“取绝对值”的函数
涉及系统调用的库函数：如“创建一个新文件”的函数



王道考研/CSKAOYAN.COM

小例子：为什么系统调用是必须的？



生活场景：去学校打印店打印论文，你按下了 WPS 的“打印”选项，打印机开始工作。
你的论文打印到一半时，另一位同学按下了 Word 的“打印”按钮，开始打印他自己的论文。



思考：如果两个进程可以随意地、并发地共享打印机资源，会发生什么情况？

两个进程并发运行，打印机设备交替地收到 WPS 和 Word 两个进程发来的打印请求，结果两篇论文的内容混杂在一起了...

解决方法：由操作系统内核对共享资源进行统一的管理，并向上提供
“系统调用”，用户进程想要使用打印机这种共享资源，只能通过系统
调用向操作系统内核发出请求。内核会对各个请求进行协调处理。



王道考研/CSKAOYAN.COM

什么功能要用到系统调用？

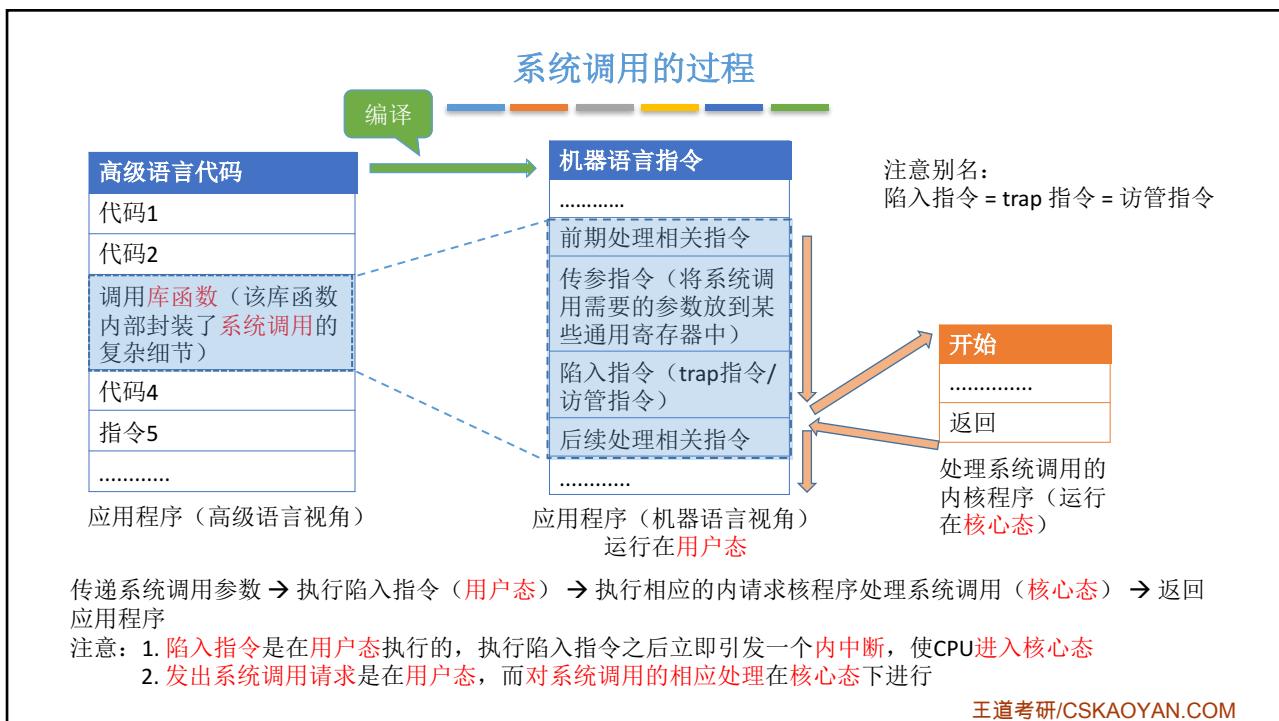
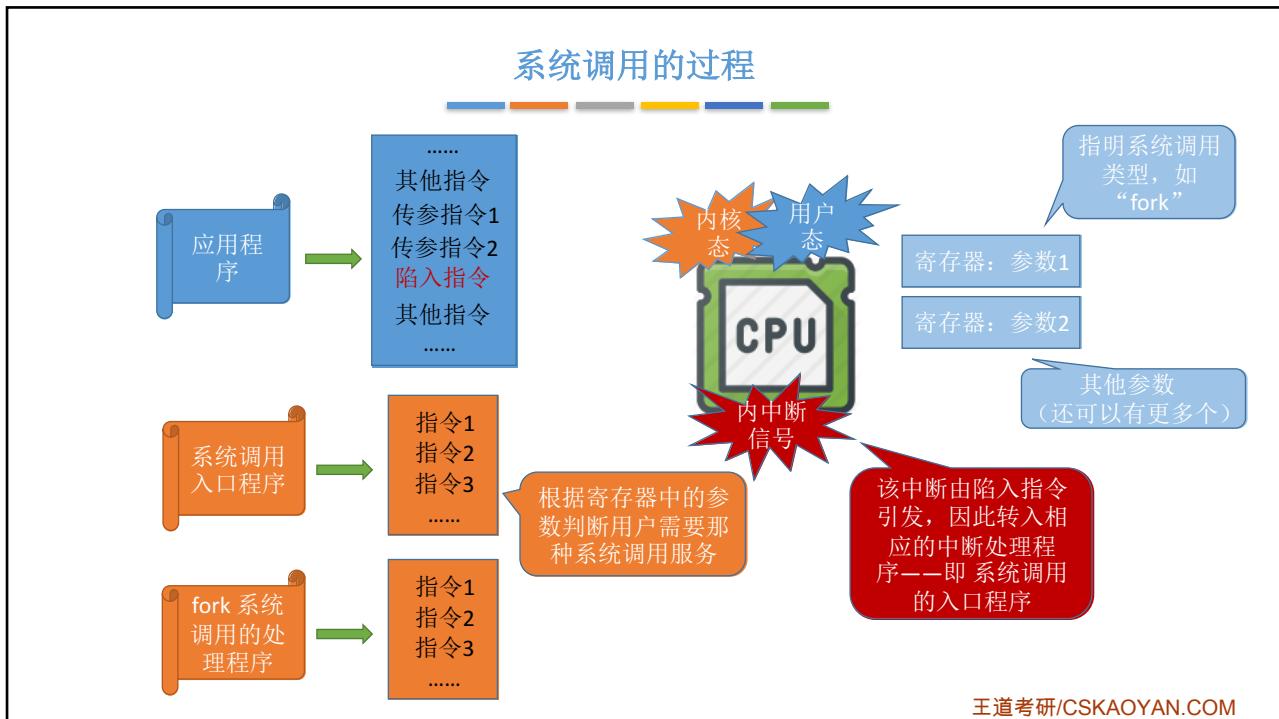
应用程序通过系统调用请求操作系统的服务。而系统中的各种共享资源都由操作系统内核统一掌管，因此凡是与共享资源有关的操作（如存储分配、I/O操作、文件管理等），都必须通过系统调用的方式向操作系统内核提出服务请求，由操作系统内核代为完成。这样可以保证系统的稳定性和安全性，防止用户进行非法操作。

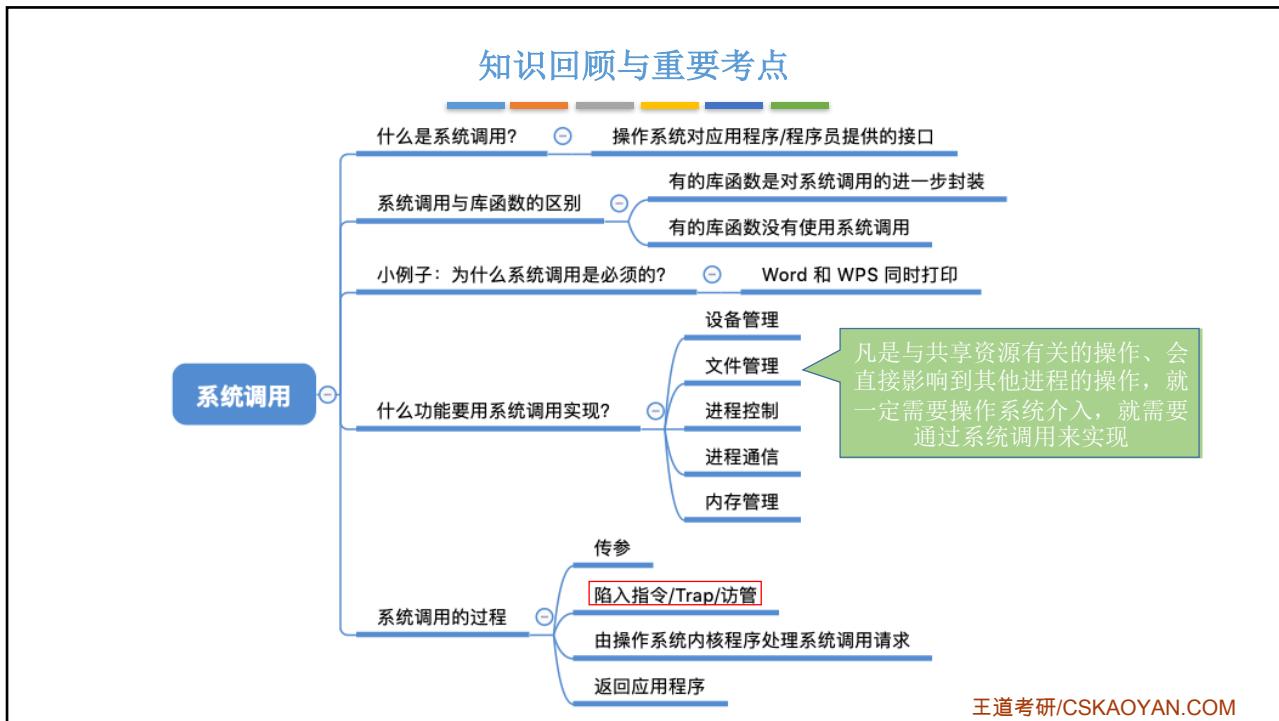
系统调用（按功能分类）



拓展：感兴趣的同学可以搜索“Linux 系统调用”，了解 Linux 操作系统提供了哪些系统调用

王道考研/CSKAOYAN.COM





本节内容

操作系统的 体系结构

王道考研/CSKAOYAN.COM

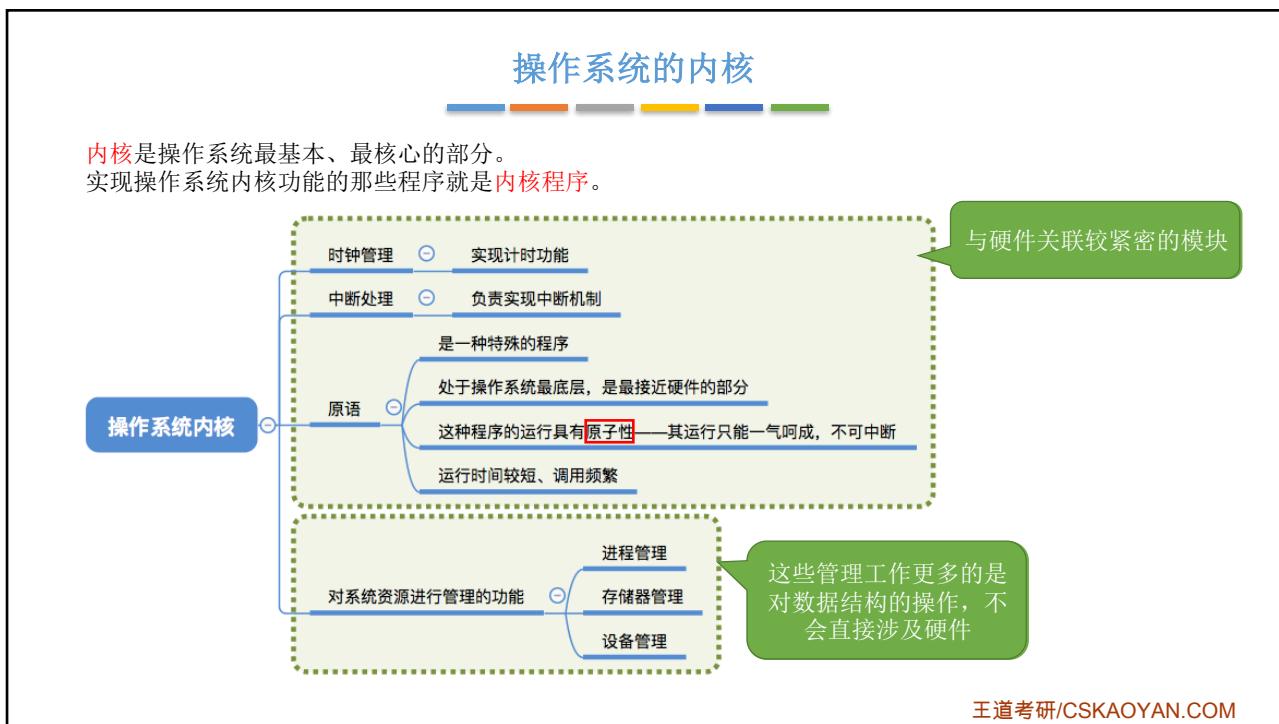
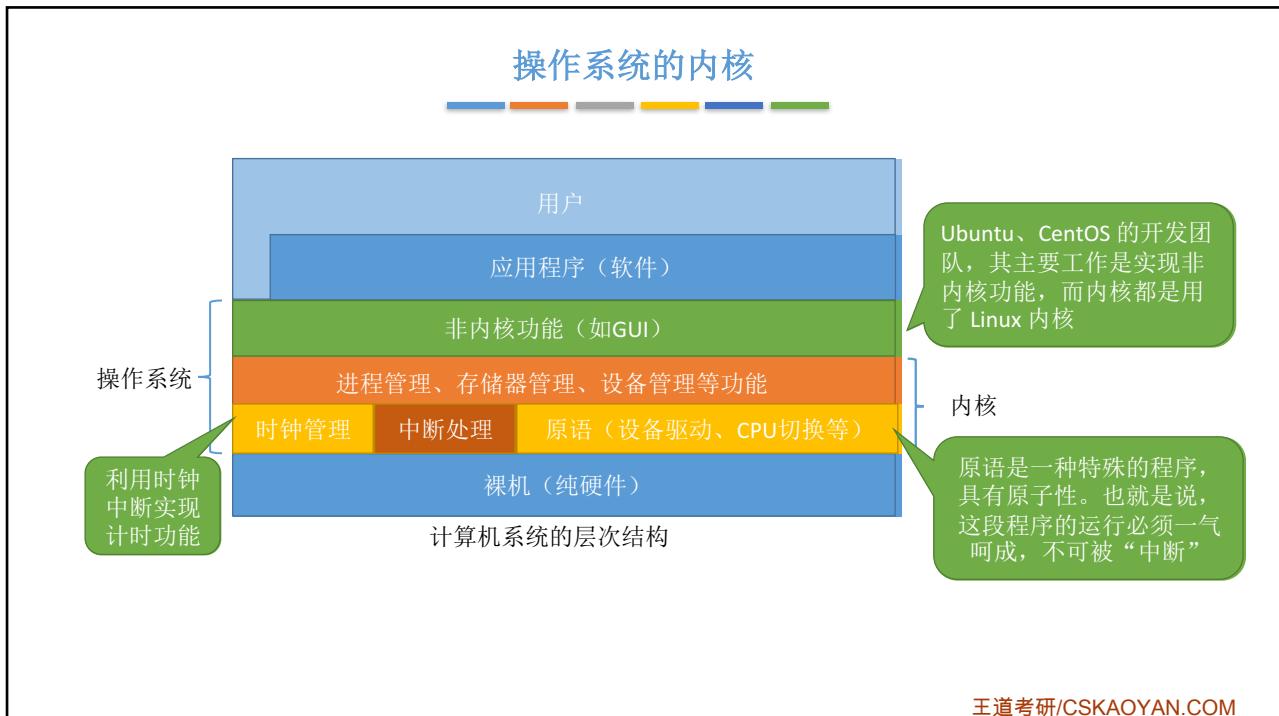
知识总览

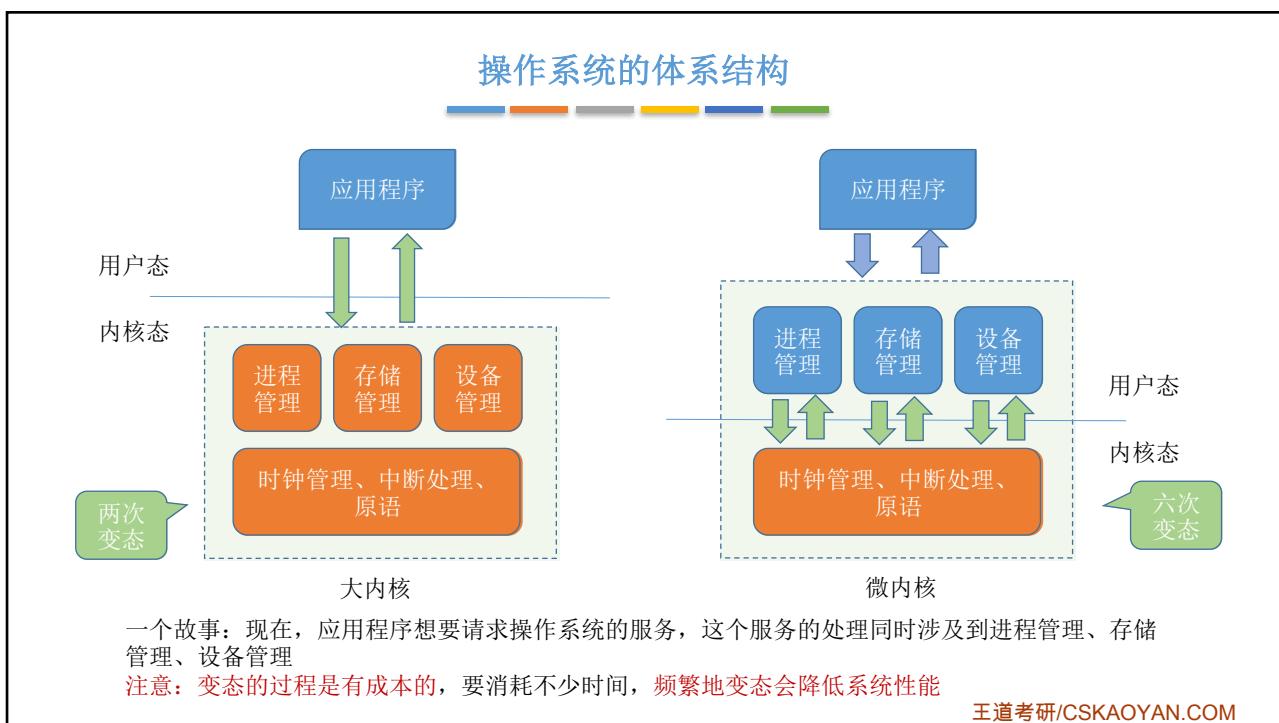
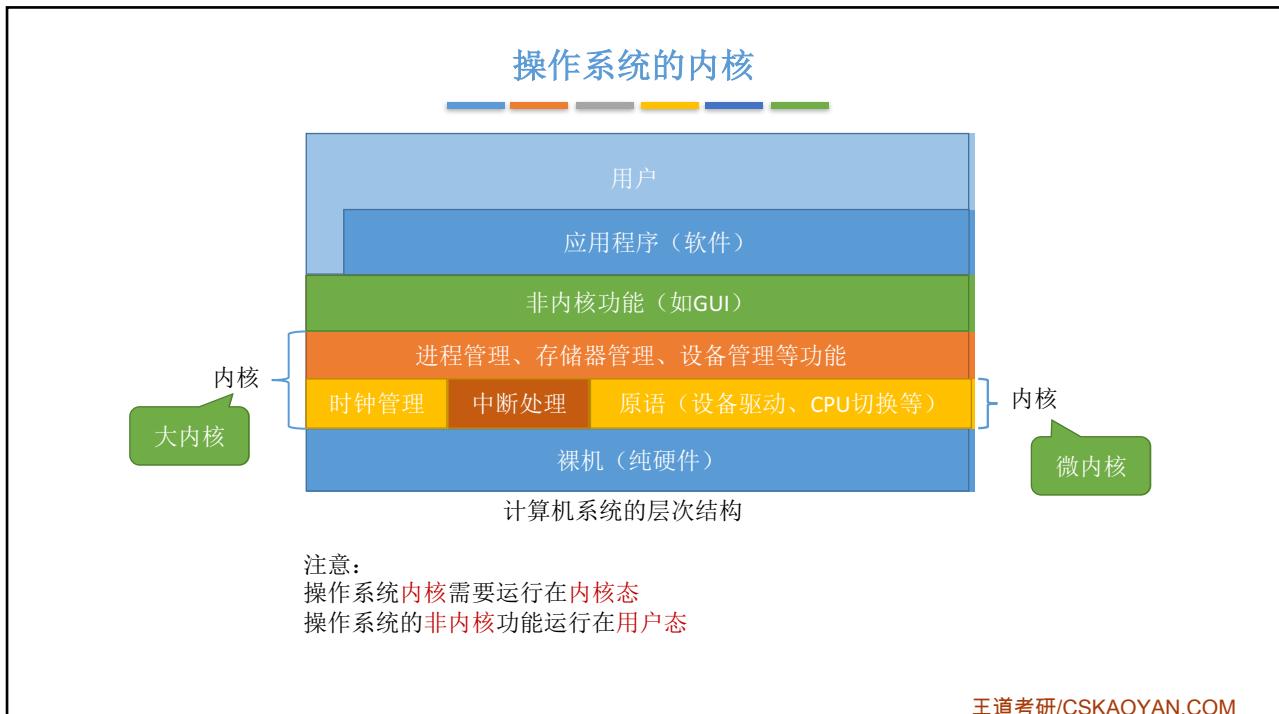
操作系统的体系结构

大内核/单内核/宏内核

微内核

王道考研/CSKAOYAN.COM





知识回顾与重要考点

The diagram illustrates the two main types of operating system kernels:

- 大内核 (Monolithic Kernel):** Described as "将操作系统的主要功能模块都作为系统内核, 运行在核心态". It has two points:
 - 优点: 高性能
 - 缺点: 内核代码庞大, 结构混乱, 难以维护
- 微内核 (Microkernel):** Described as "只把最基本的功能保留在内核". It also has two points:
 - 优点: 内核功能少, 结构清晰, 方便维护
 - 缺点: 需要频繁地在核心态和用户态之间切换, 性能低

操作系统的体系结构

典型的大内核/宏内核/单内核 操作系统: Linux、UNIX
典型的 微内核 操作系统: Windows NT

王道考研/CSKAOYAN.COM

