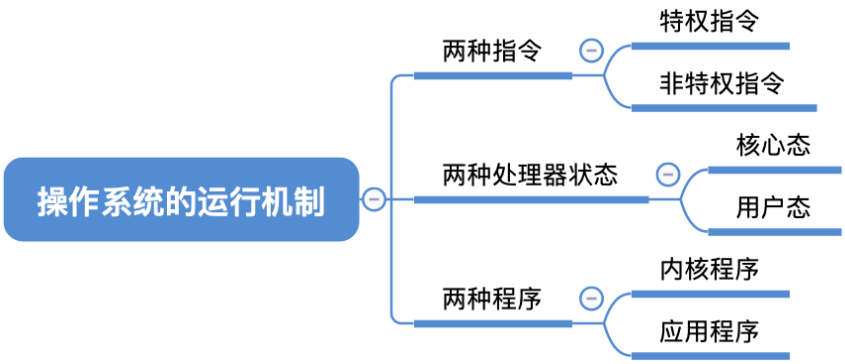
本节内容

操作系统的运行机制

知识总览



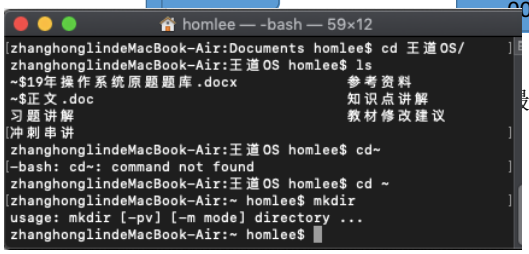
预备知识：程序是如何运行的？

C语言代码 → 编译器（“翻译”） → 机器指令（二进制）

一条高级语言的代码翻译过来可能会对应多条机器指令

Int x = 1;
x++;

```
10001010110000110001110
00101100000000110100111
10010010000000110000000
00101100010011110000001
```



基本命令

“小黑框”中使用的命令也
注意与本节的“指令”区别

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

预备知识：程序是如何运行的？

C语言代码 → 编译器（“翻译”） → 机器指令（二进制）

一条高级语言的代码翻译过来可能会对应多条机器指令

Int x = 1;
x++;

```
10001010110000110001110
00101100000000110100111
10010010000000110000000
00101100010011110000001
```

“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

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

王道考研/CSKAOYAN.COM

内核程序 v.s. 应用程序

C语言代码 → 编译器（“翻译”） → 机器指令（二进制）

一条高级语言的代码翻译过来可能会对应多条机器指令

Int x = 1;
x++;

```
10001010110000110001110
00101100000000110100111
10010010000000110000001
00101100010011110000011
```

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

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

Int x = 1;
x++;

```
10001010110000110001110
00101100000000110100111
10010010000000110000001
00101100010011110000011
```

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

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”

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

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

王道考研/CSKAOYAN.COM

内核态 v.s. 用户态

Int x = 1;
x++;

```
10001010110000011000011100
00101100000000011010011111
10010010000000011000000001
00101100010011110000011
```

CPU

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

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？

问题：如何实现CPU状态的切换？

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

内核程序

应用程序

```
00101100000000011010011111
10010010000000011000000001
10001010110000011000011100
00101100010011110000011
```

内核态

用户态

CPU

中断信号

处理中断信号的内核程序

```
00000100101111110111010
10010101000101110111010
00101001001010001010010
```

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”

① [illegible]为“内核态”，
② [illegible]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

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夺回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

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”

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

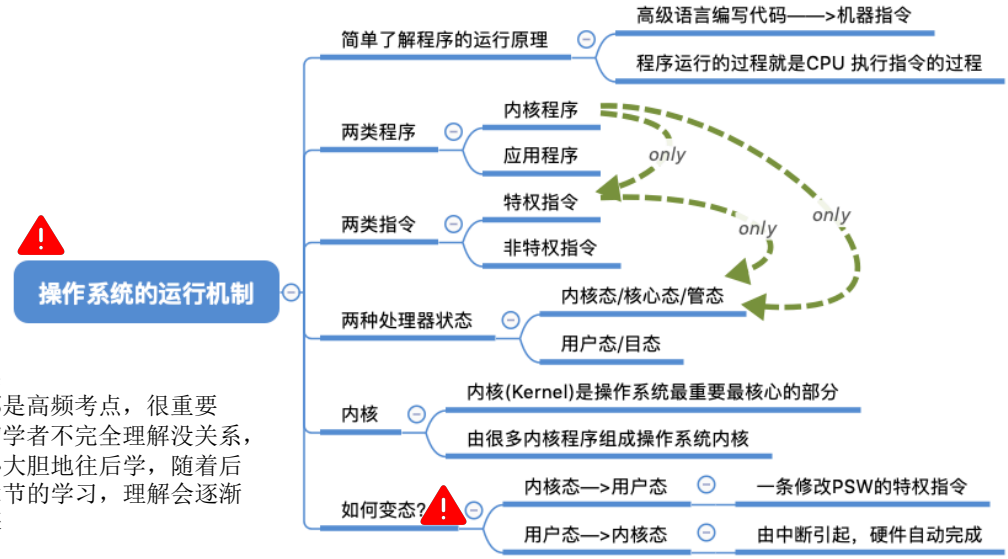
CPU检测到中断信号后，会立即变为“核心态”，并停止运行当前的应用程序，转而运行处理中断信号的内核程序

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

知识回顾与重要考点



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

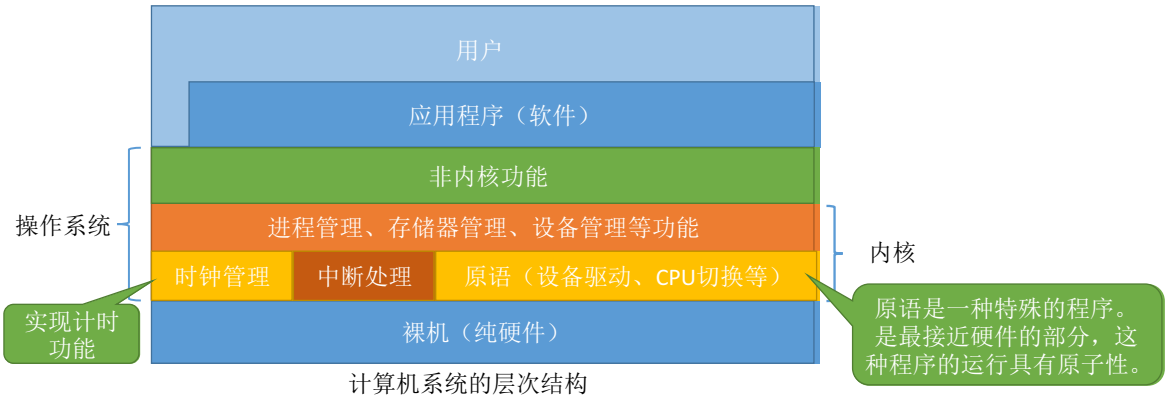
两种指令、两种处理器状态、两种程序

两种程序
- 内核程序：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。
- 应用程序：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

王道考研/CSKAOYAN.COM

操作系统的内核

Yo~生活经验：我们安装完 Windows 操作系统后，会发现操作系统提供了多种多样的功能，比如“记事本”、“任务管理器”。然而，这些功能并不是必不可少的。即使没有“任务管理器”，我们仍然可以使用计算机。

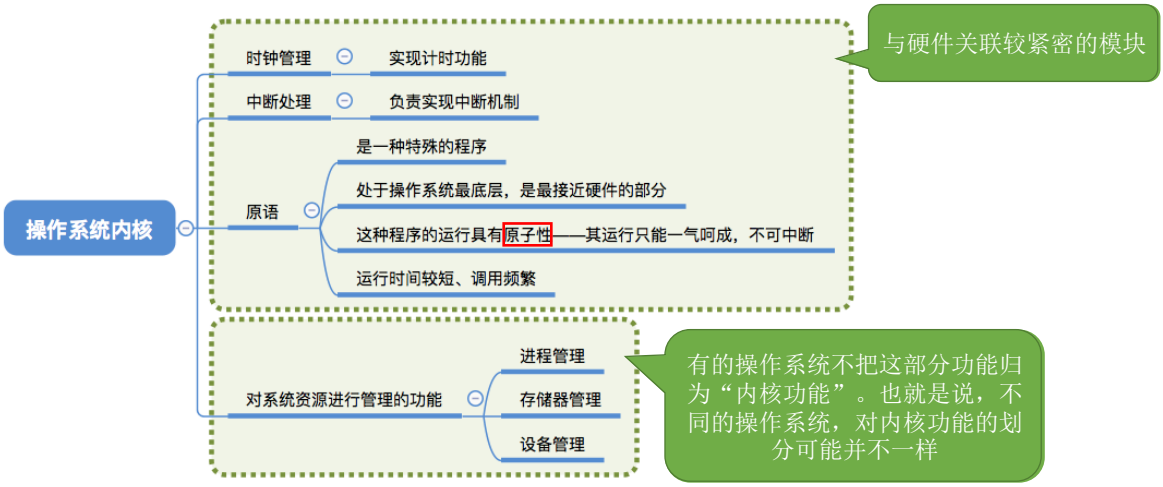


计算机系统的层次结构

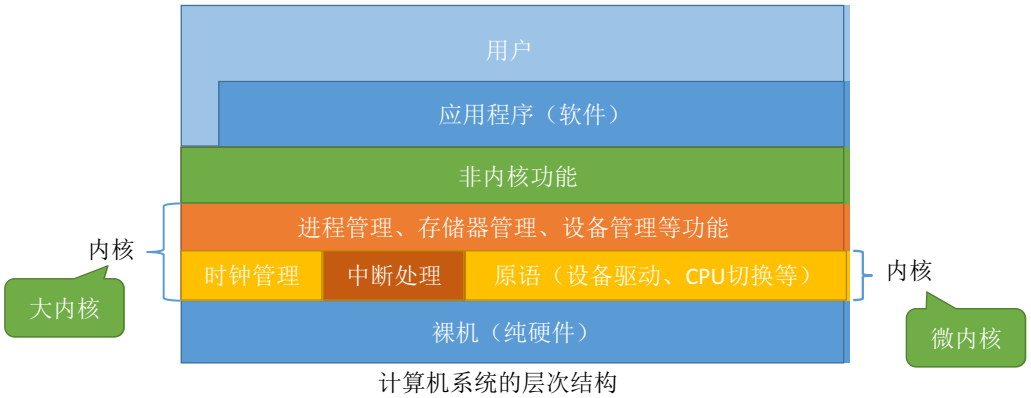
王道考研/CSKAOYAN.COM

操作系统的内核

内核是计算机上配置的底层软件，是操作系统最基本、最核心的部分。
实现操作系统内核功能的那些程序就是内核程序。



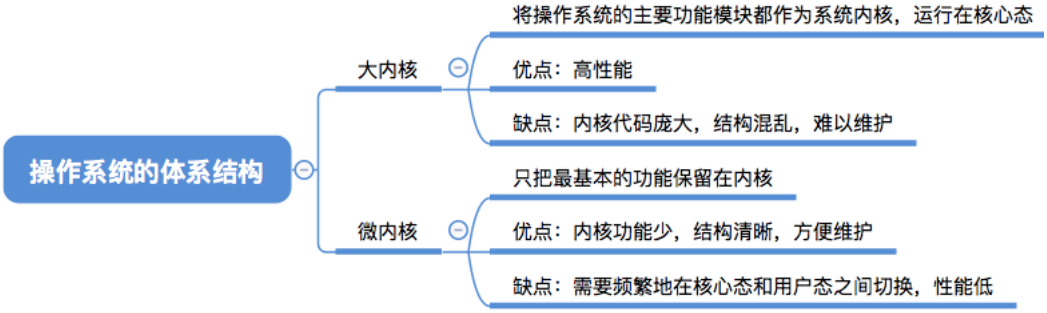
操作系统的内核



计算机系统的层次结构

操作系统的体系结构：大内核和微内核

操作系统的体系结构



类比：
操作系统的体系结构问题与企业的管理问题很相似。
内核就是企业的管理层，负责一些重要的工作。只有管理层才能执行特权指令，普通员工只能执行非特权指令。用户态、核心态之间的切换相当于普通员工和管理层之间的工作交接
大内核：企业初创时体量不大，管理层的人会负责大部分的事情。优点是效率高；缺点是组织结构混乱，难以维护。
微内核：随着企业体量越来越大，管理层只负责最核心的一些工作。优点是组织结构清晰，方便维护；缺点是效率低。

王道考研/CSKAOYAN.COM

知识回顾与重要考点

最常考知识点：
1. 特权指令只能在核心态下执行
2. 内核程序只能在核心态下执行
3. 核心态、用户态之间的切换（后续讲解内容）

王道考研/CSKAOYAN.COM