计算机考研系列书课包

# 玩转操作系统

主讲人 刘财政

# 第一讲　操作系统概述

# 操作系统与我们永在







# 本讲内容

考点一：操作系统的概念

考点二：操作系统的发展历程 ★★★

考点三：操作系统运行环境 ★★★★★

考点四：操作系统的结构 ★

考点五： 虚拟机

# 考点三：操作系统运行环境

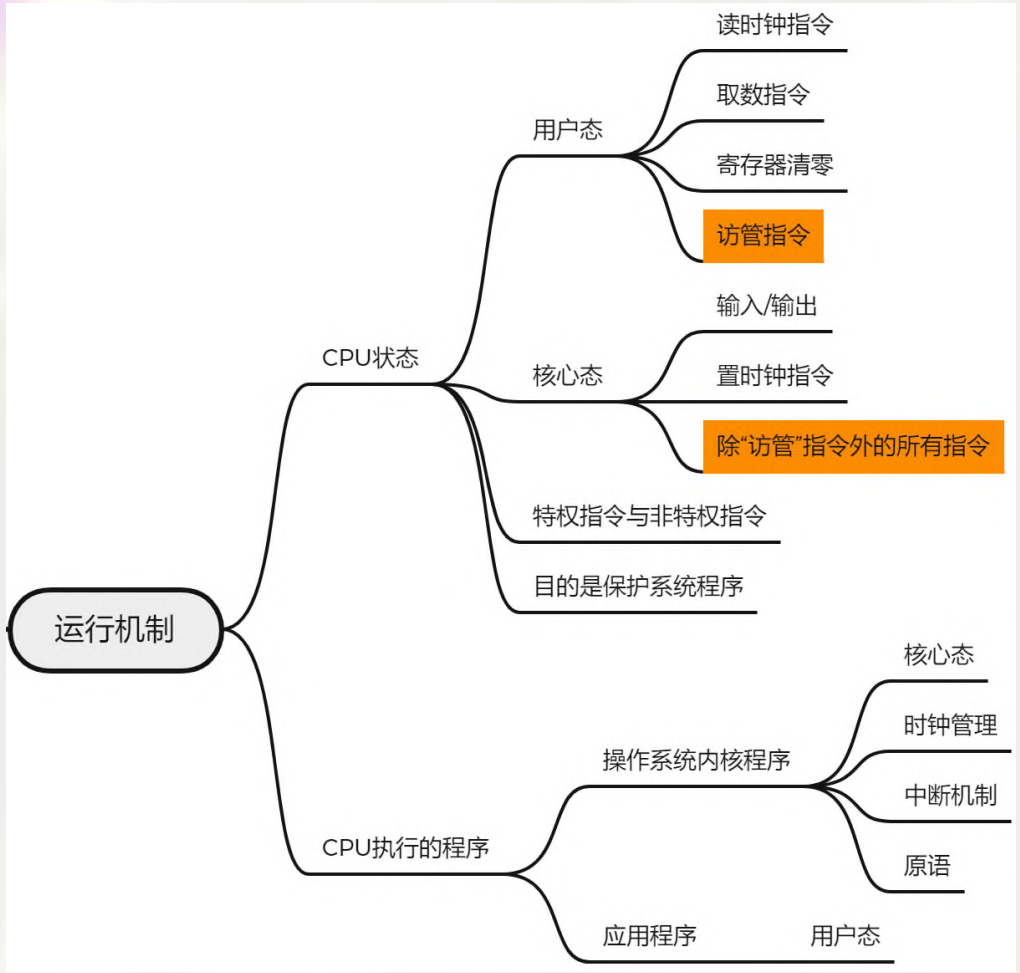
# 考点框架

- 操作系统的运行机制
- 中断
- 系统调用

# 考点三：操作系统运行环境

## 操作系统的运行机制

# 考点三：操作系统运行环境




三个二：

- 二种状态
- 二类指令
- 二类程序

# 考点三：操作系统运行环境

## 程序的运行过程

C语言代码 —编译器（“翻译”）→ 机器指令

```
int x = 1;
x++;
```

```
10001010110000110001110000101100000000011010011111001001000000011000000001001011000100111100000011
```

“指令”就是处理器（CPU）能识别、执行的最基本命令

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

# 考点三：操作系统运行环境

## 两类程序

```
int x = 1;
x++;
```

```
10001010110000110001110000
10110000000011010011110010
01000000110000000100101100
01001111000000110010110001
```

我们普通程序员写的程序就是“应用程序”

支付宝 - 让生活更简单

CPU

# 考点三：操作系统运行环境

## 两类程序

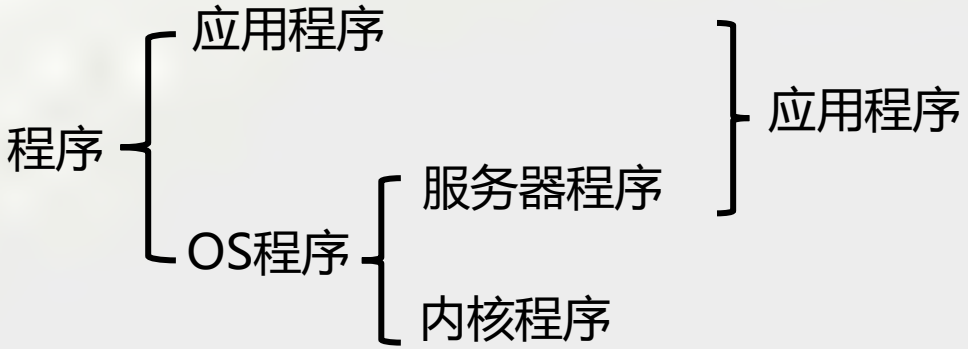
```
int x = 1;
x++;
```

```
10001010110000110001110000
00101100000000011010011111
10010010000000110000000001
00101100010011110000000011
```

- 微软、linux社区有一帮人负责实现操作系统，他们写的是“内核程序”组成了“操作系统内核”，或简称“内核”
- 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
- 操作系统的功能未必都在内核中，

# 考点三：操作系统运行环境

## 两类程序




对操作系统而言，这两种程序的作用不同，内核程序是应用程序的管理者，因此内核程序(管理程序）要执行一些特权指令，而用户自编程序 (被管理程序）出于安全考虑不能执行这些指令。

# 考点三：操作系统运行环境

## 两种状态

- CPU 有两种状态，内核态和用户态。多数操作系统将处理器工作状态划分为内核态和用户态。
- 内核态一般指操作系统内核程序运行的状态，具有较高的特权级别，又称为特权态、系统态或管态；
- 用户态一般指用户程序运行时的状态，具有较低的特权级别，又称为普通态、目态，常态

# 考点三：操作系统运行环境

## 两种状态

- 目态：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，是用户程序运行时的状态，较低的特权级别。
- 管态：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。执行系统管理程序。程序执行时可使用特权指令，I/O 指令、时钟设置、中断机制、系统管理等，
- 用户态不能干扰内核态.

# 考点三：操作系统运行环境

## 两种状态



- CPU的状态寄存器（PSWR）中记录了当前CPU的工作状态,其中有一个二进制位，1表示"核心态"，0表示"用户态"。
- 工作状态就是指：**CPU当前是工作在内核态还是用户态**

## 考点三：操作系统运行环境

### 两种状态

用户态向内核态的转换

- **系统调用**：这是用户态进程主动要求切换到内核态的一种方式，用户态进程通过系统调用申请使用操作系统提供的服务程序完成工作，而系统调用的机制其核心还是使用了操作系统为用户特别开放的一个中断来实现。

## 考点三：操作系统运行环境

### 两种状态

用户态向内核态的转换

- **异常**：当CPU在执行运行在用户态下的程序时，发生了某些事先不可知的异常，这时会触发由当前运行进程切换到处理此异常的内核相关程序中，也就转到了内核态，比如缺页异常。

## 考点三：操作系统运行环境

### 两种状态

用户态向内核态的转换:

- **外围设备的中断**：当外围设备完成用户请求的操作后，会向CPU发出相应的中断信号，这时CPU会暂停执行下一条即将要执行的指令转而去执行与中断信号对应的处理程序，如果先前执行的指令是用户态下的程序，那么这个转换的过程自然也就发生了由用户态到内核态的切换。

## 考点三：操作系统运行环境

### 两种状态

而内核态到用户态的切换比较简单，内核程序执行一条特权指令，即修改PSW的标志位用户态，这个动作意味着操作系统将主动让出CPU的使用权。

# 考点三：操作系统运行环境

## 两类指令

CPU的两种状态相对应的是两类指令，即特权指令和非特权指令。

- 特权指令：是指计算机中不允许用户直接使用的指令，如 I/O指令、置中断指令，存取用于内存保护的寄存器、送程序状态字到程序状态字寄存器等的指令（这几种记好，属于内核态），说明此时正在运行的是内核程序，此时可以执行特权指令。
- 非特权指令：“管理程序”(即用户自编程序）出于安全考虑不能执行的指令，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。应用程序只能使用“非特权指令”，如：加法指令、减法指令等。

# 考点三：操作系统运行环境

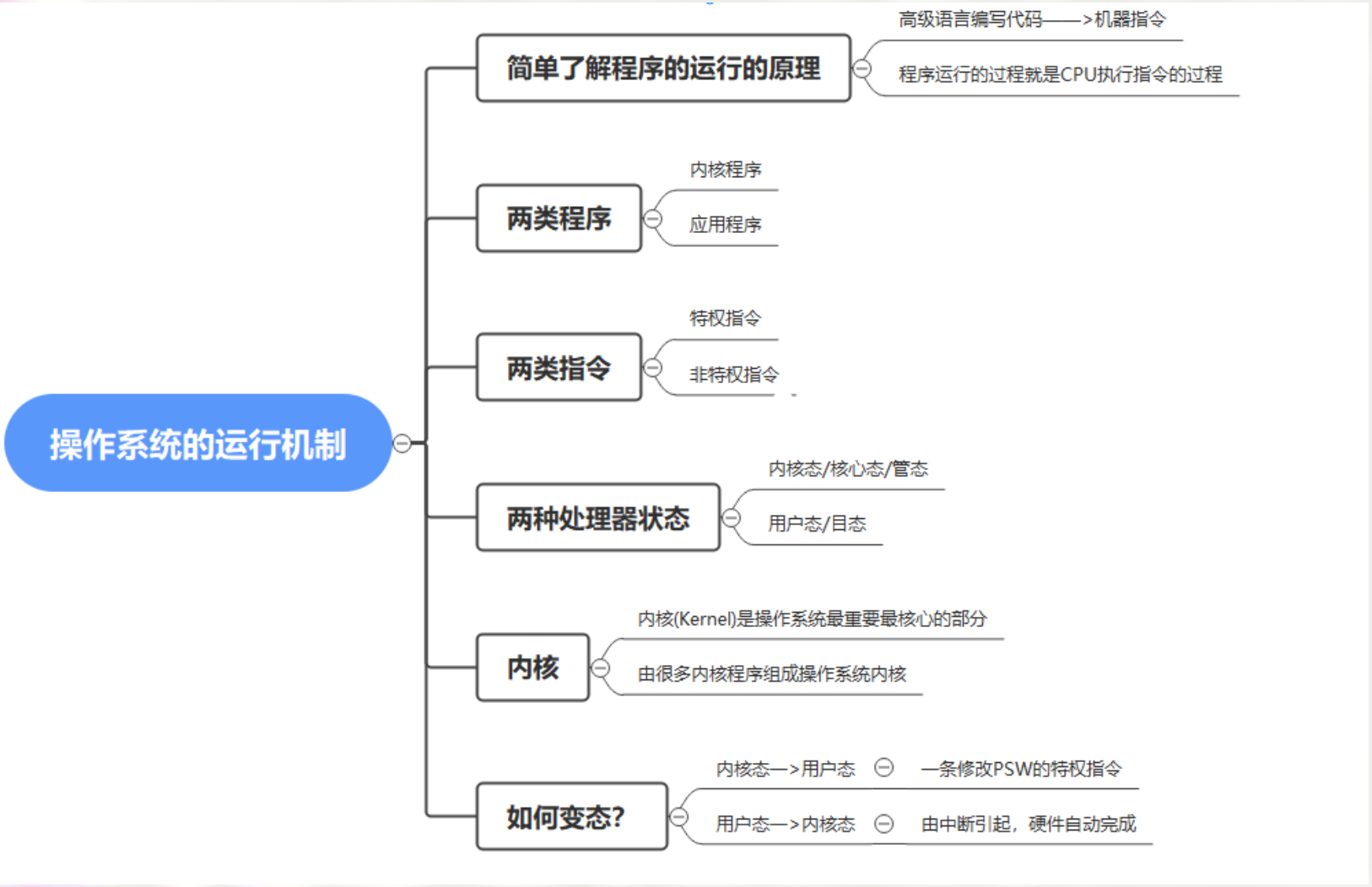
## 两类指令

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

| Bit # | 31..26 | 25..21 | 20..16 | 15..0 | |
|---|---|---|---|---|---|
| **I-type** | **op** | **rs** | **rt** | **immediate** | |
| addi | 001000 | rs | rt | Immediate(- ~ +) | addi $1,$2,100 |
| addiu | 001001 | rs | rt | Immediate(- ~ +) | addiu $1,$2,100 |
| andi | 001100 | rs | rt | Immediate(0 ~ +) | andi $1,$2,10 |
| ori | 001101 | rs | rt | Immediate(0 ~ +) | andi $1,$2,10 |
| xori | 001110 | rs | rt | Immediate(0 ~ +) | andi $1,$2,10 |
| lw | 100011 | rs | rt | Immediate(- ~ +) | lw $1,10($2) |
| sw | 101011 | rs | rt | Immediate(- ~ +) | sw $1,10($2) |
| beq | 000100 | rs | rt | Immediate(- ~ +) | beq $1,$2,10 |
| bne | 000101 | rs | rt | Immediate(- ~ +) | bne $1,$2,10 |
| slti | 001010 | rs | rt | Immediate(- ~ +) | slti $1,$2,10 |
| sltiu | 001011 | rs | rt | Immediate(- ~ +) | sltiu $1,$2,10 |
| lui | 001111 | 00000 | rt | Immediate(- ~ +) | Lui $1, 10 |

| Bit # | 31..26 | 25..0 | |
|---|---|---|---|
| **J-type** | **op** | **Index** | |
| j | 000010 | address | j 10000 |
| jal | 000011 | address | jal 10000 |

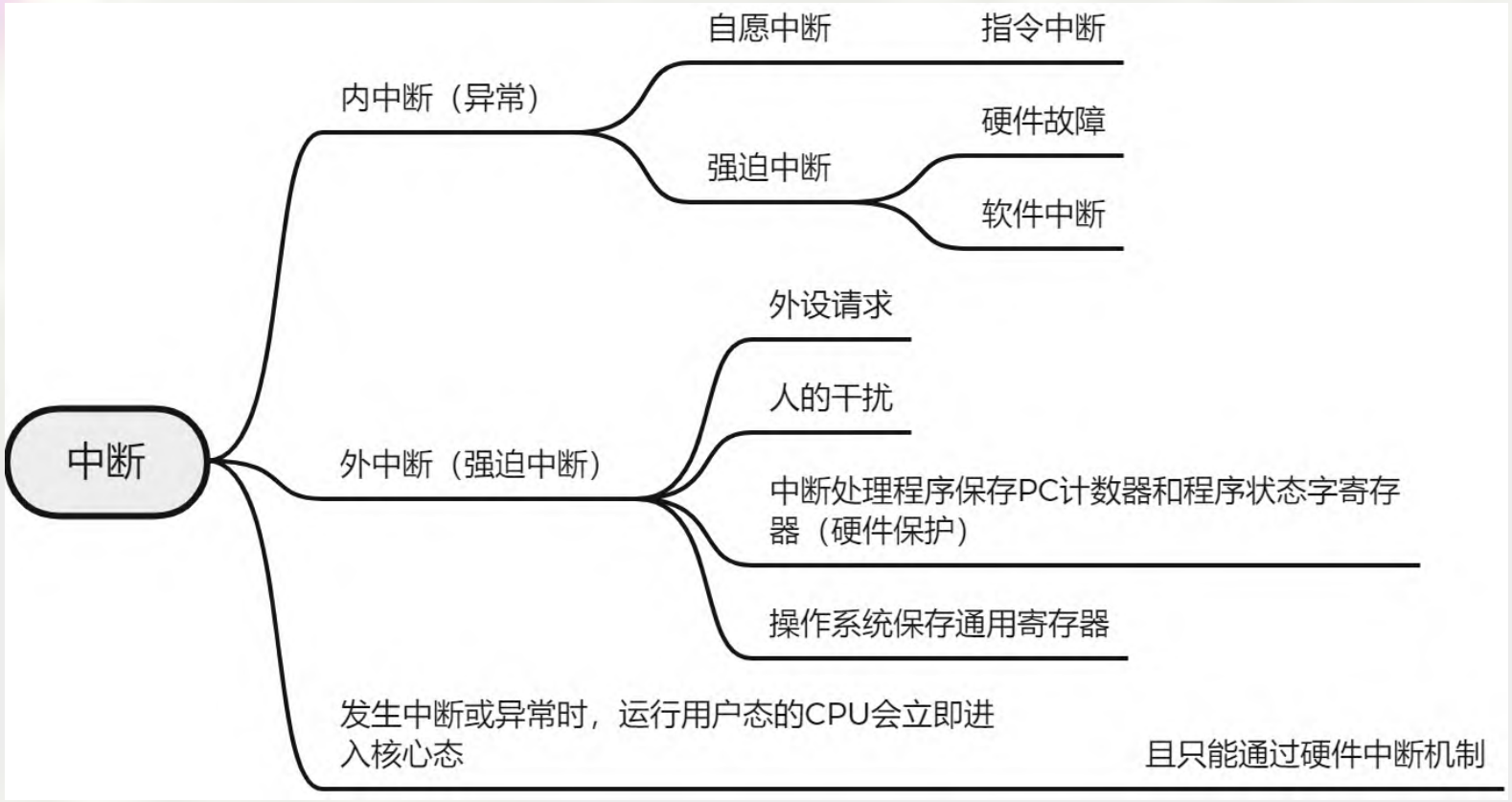
# 考点三：操作系统运行环境



# 考点三：操作系统运行环境

## 中断

# 考点三：操作系统运行环境




## 【与计算机组成原理融合】

# 考点三：操作系统运行环境

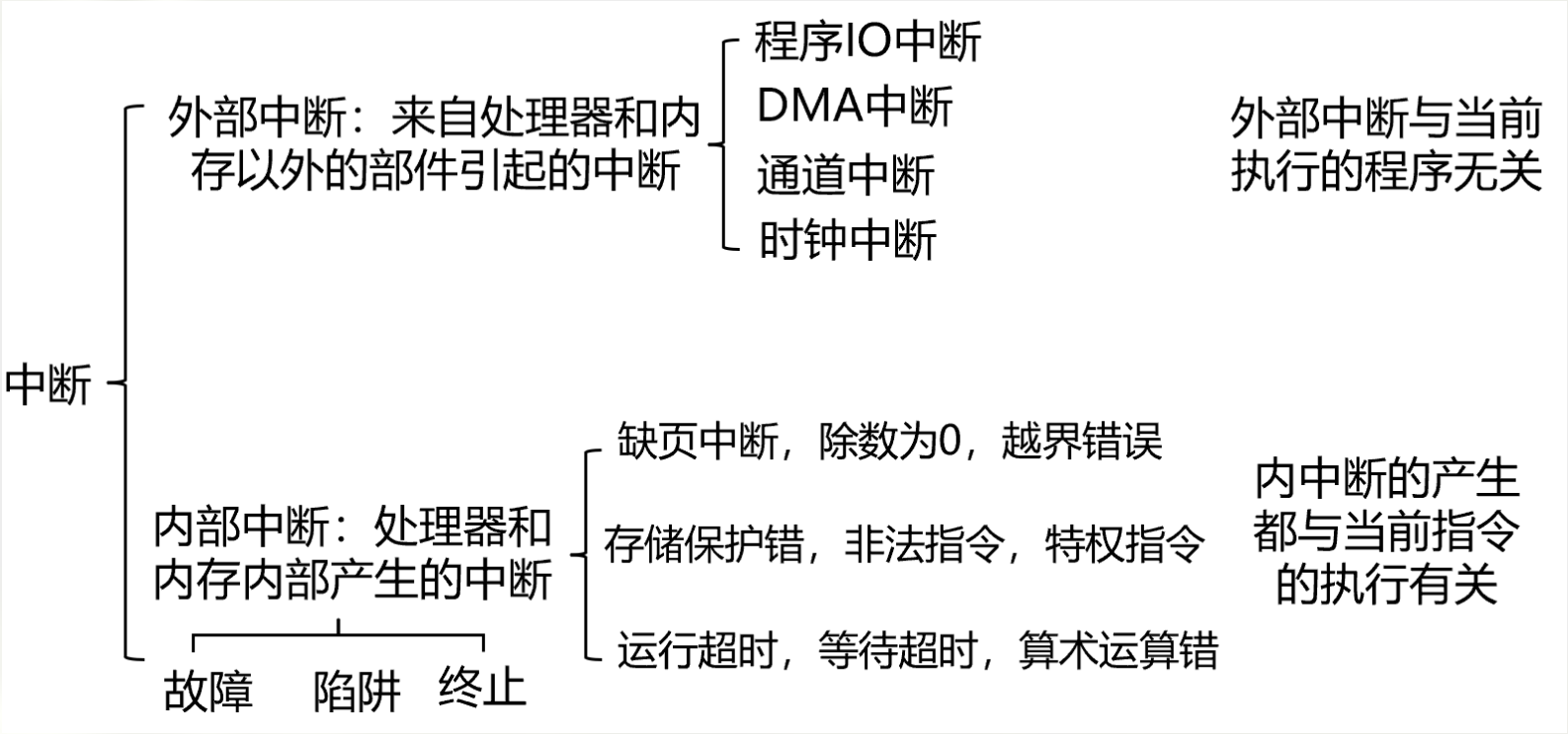
## 中断的概念

在CPU执行程序的过程中，出现了某种紧急或异常的事件（内部异常或外部中断等原因），CPU需暂停正在执行的程序，转去处理该事件（执行中断服务程序），并在处理完毕后返回断点处继续执行被暂停的程序，这一过程称为中断。

# 考点三：操作系统运行环境

## 中断的分类

- 按照中断源的位置，可以分为内部中断和外部中断



# 考点三：操作系统运行环境

## 故障

- 故障是引起故障的指令在执行过程中CPU检测到的一类与指令执行相关的意外事件。这种意外事件有些可以恢复，有些则不能恢复。

  例如，指令译码时出现“非法操作码”；取指令或数据时发生“页故障（page fault）”；执行除法指令时发现“除数为0”等。

# 考点三：操作系统运行环境

## 故障

- 溢出和非法操作码等这类故障，因为无法通过异常处理程序恢复所以不能回到被中断的程序继续执行，
- 通常异常处理程序在屏幕上显示一个对话框告知发生了某种故障，然后调用内核中的abort例程,以终止发生故障的当前进程。

# 考点三：操作系统运行环境

## 故障

- 对于除数为0的情况，根据是定点除法指令还是浮点除法指令有不同的处理方式。
  - 对于浮点数除0，异常处理程序可以选择将指令执行结果用特殊的值（如α或NaN表示，然后返回到用户进程继续执行除法指令后面的一条指令;
  - 对于整数除0，则会发生“整除0”故障,通常调用abort 例程来终止当前用户进程。

# 考点三：操作系统运行环境

## 故障

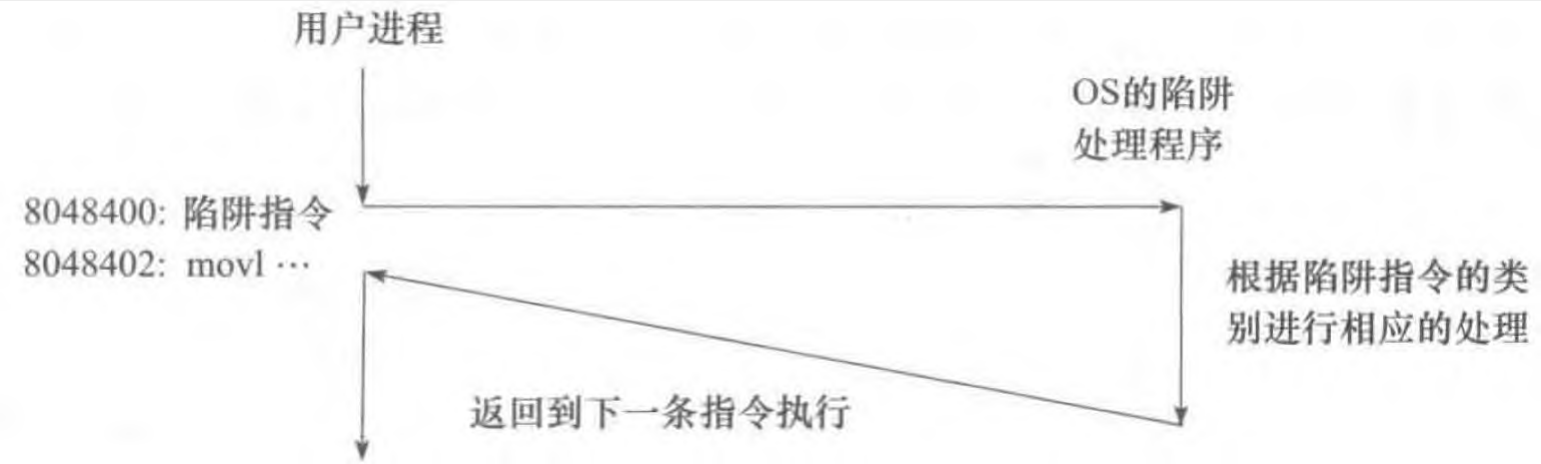
对于页故障，对应的页故障处理程序会根据不同的情况进行不同的处理。

- 首先检测是否发生地址越界或访问越权，如果是的话，则故障不可恢复；否则是真正的缺页故障，此时，可以通过从硬盘读入页面来恢复故障。

# 考点三：操作系统运行环境

## 陷阱

- 陷阱也称为自陷或陷入，是预先安排的一种“异常”事件，就像预先设定的“陷阱”一样。
- 当执行到陷阱指令（也称为自陷指令）时,CPU就调出特定的程序进行相应的处理，处理结束后返回到陷阱指令的下一条指令执行。



## 考点三：操作系统运行环境

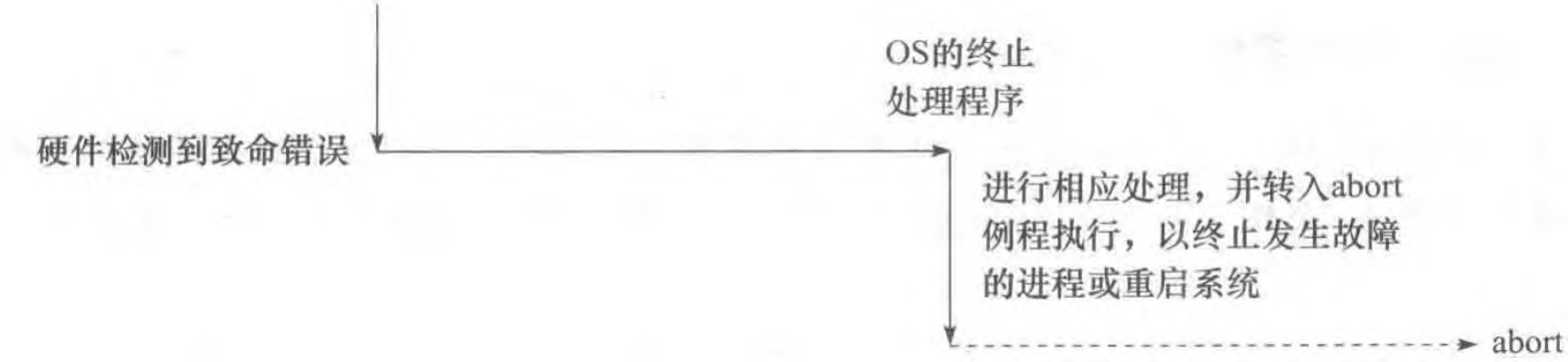
### 陷阱

- 陷阱的重要作用之一是在用户程序和内核之间提供一个像过程一样的接口，这个接口称为系统调用，用户程序利用这个接口可以方便地使用操作系统内核提供的一些服务。
- 操作系统给每个服务编一个号，称为系统调用号，每个服务功能通过一个对应的系统调用服务例程提供。

# 考点三：操作系统运行环境

## 终止

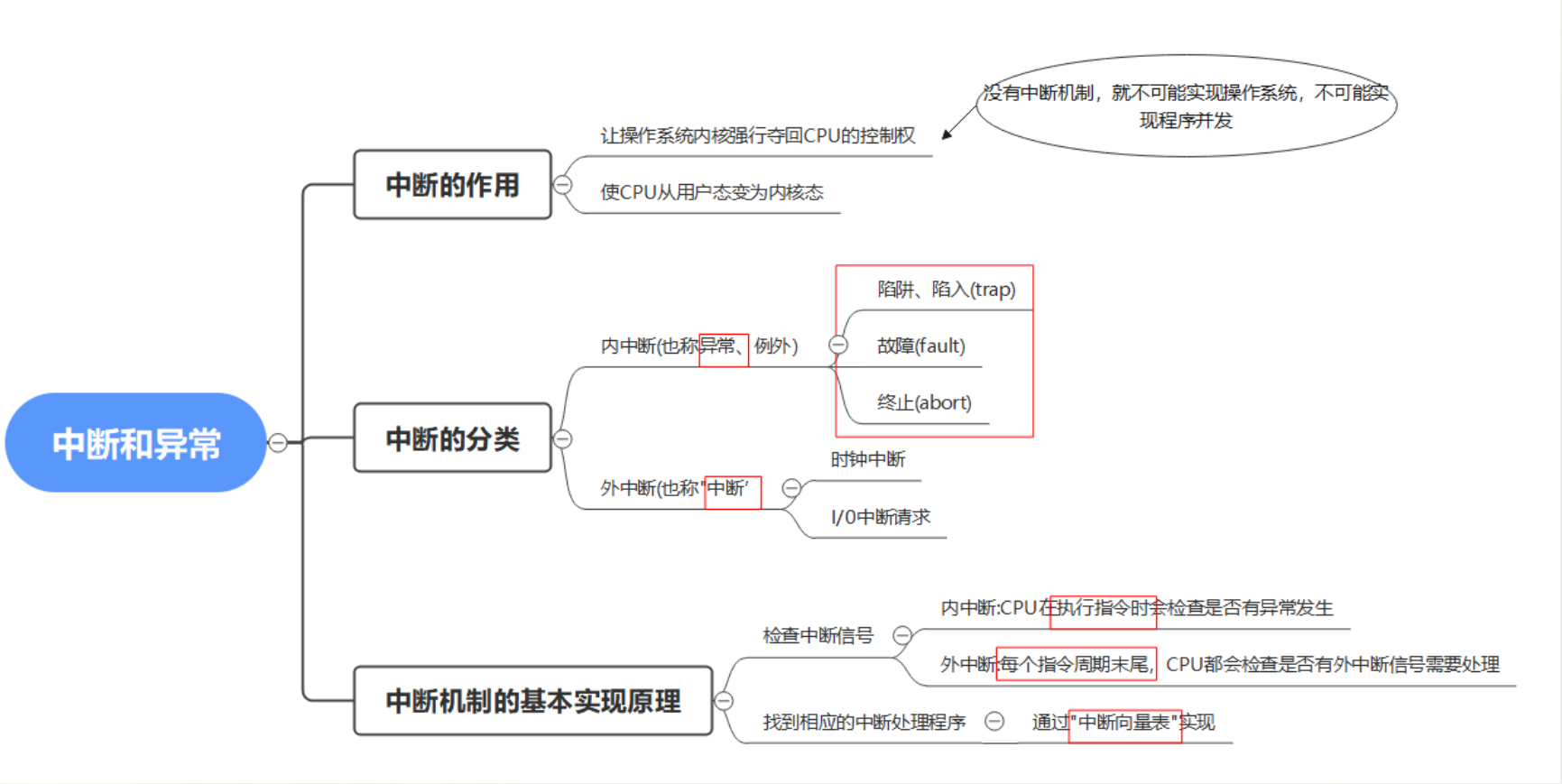
如果在执行指令过程中发生了严重错误，例如，控制器出现问题，访问DRAM或SRAM时发生校验错等，则程序将无法继续执行，只好终止发生问题的进程，在有些严重的情况下，甚至要重启系统。显然，这种异常是随机发生的，无法确定发生异常的是哪条指令。



# 考点三：操作系统运行环境

## 系统调用

# 考点三：操作系统运行环境



# 考点三：操作系统运行环境

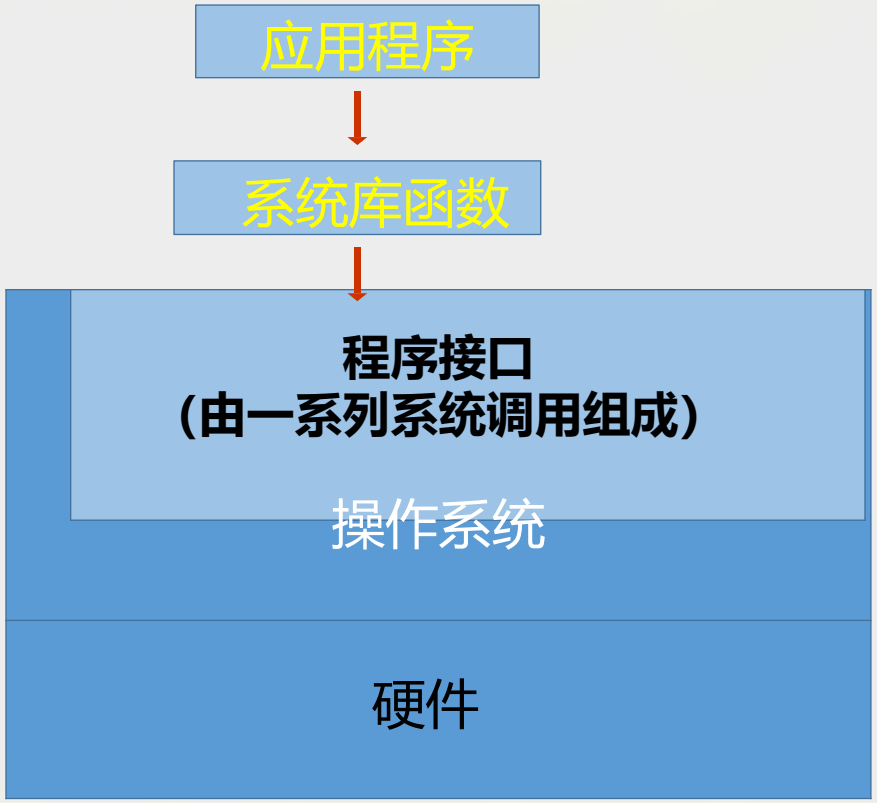
## 系统调用概念


- 系统调用(system call）提供了操作系统提供的有效服务界面。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

# 考点三：操作系统运行环境

## 系统调用

- 系统调用方式：OS提供了一组系统调用，用户可在自己的应用程序中通过相应的系统调用，来操纵计算机。



# 考点三：操作系统运行环境

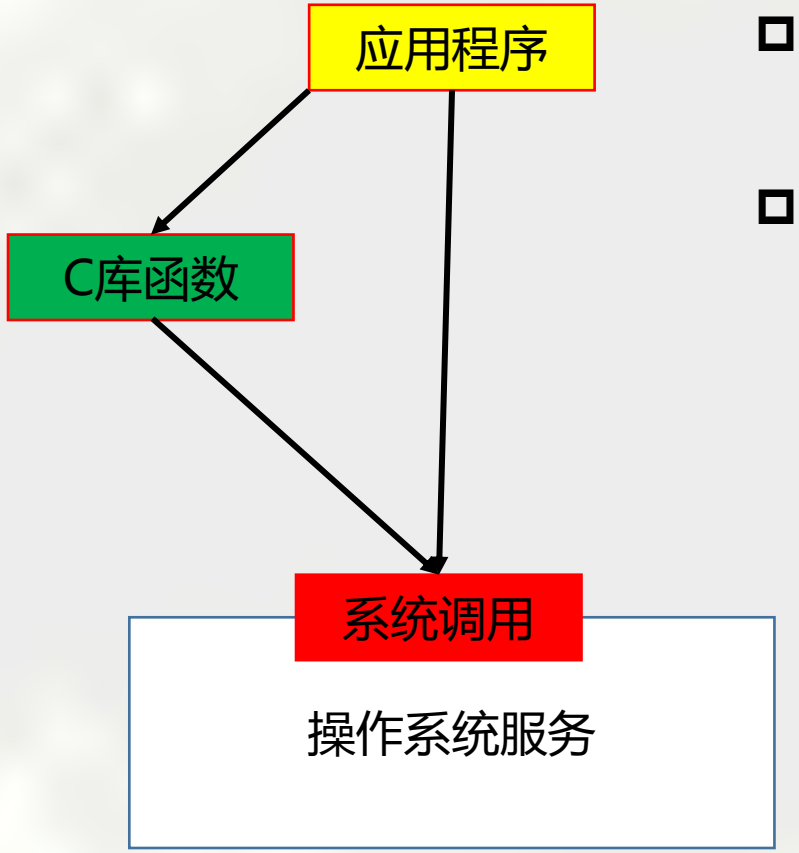
## 系统调用

- 操作系统如何使其系统调用可用之前，首先用一个例子来解释如何使用系统调用: 编写一个从一个文件读取数据并复制到另一个文件的简单程序。


系统调用顺序例子
获取输入文件名
    屏幕输入提示
    接收输入
获取输出文件名
    屏幕输入提示
    接收输入
打开输入文件
    如果文件不存在，提示失败
创建输出文件
    如果文件不存在，提示失败
循环
    读取输入文件
    写入输出文件
直到读取失败
关闭输出文件
将完成信息输输出到屏幕
正常结束

# 考点三：操作系统运行环境

## 系统调用




- 一般应用程序开发人员根据应用程序接口(API）设计程序。
- API是一系列适用于应用程序员的函数，包括传递给每个函数的参数及其返回的程序员想得到的值。

# 考点三：操作系统运行环境

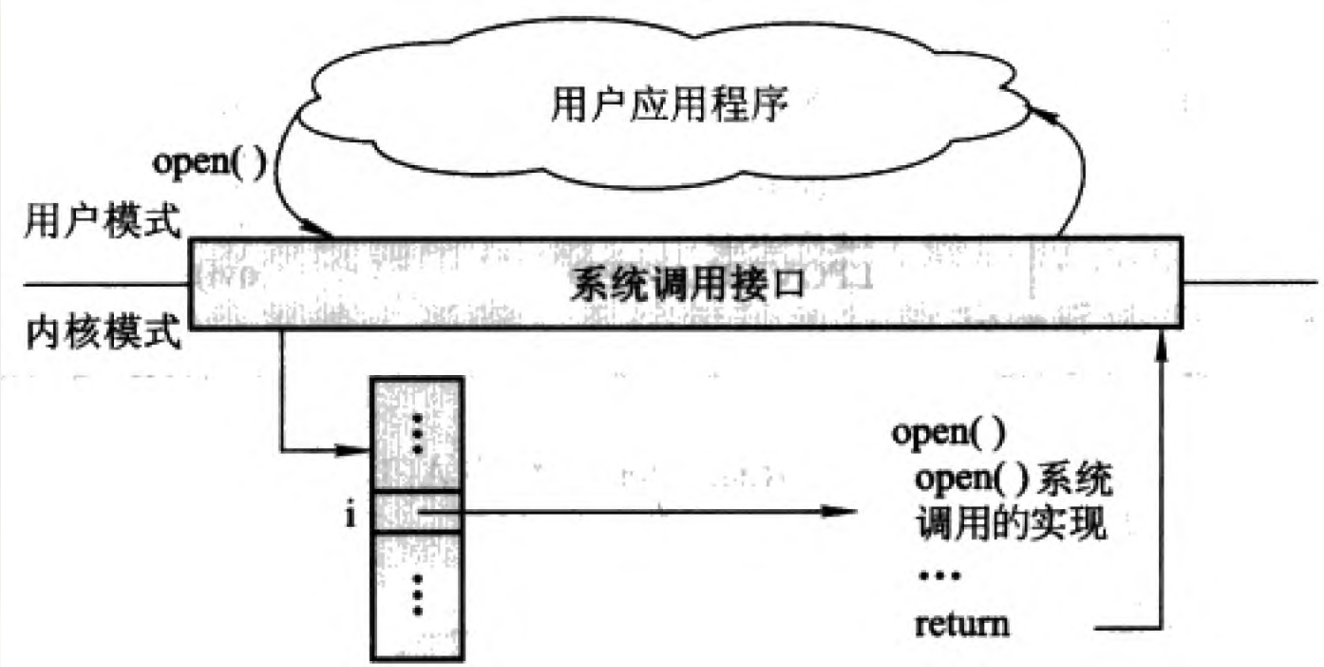
## 系统调用

- 调用者不需要知道如何执行系统调用或者执行过程，它只需遵循API并了解执行系统调用后，系统给到的结果;
- 对于程序员，通过API操作系统接口的绝大多数细节被隐藏起来，并被执行支持库所管理。

# 考点三：操作系统运行环境

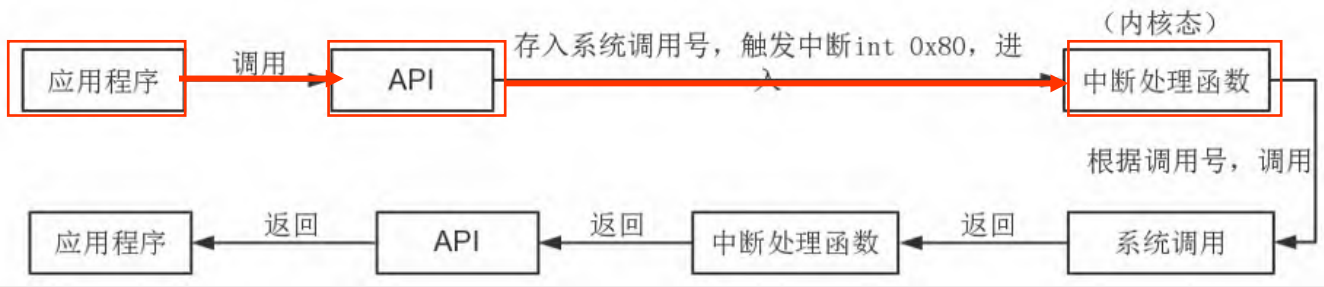
## 系统调用

API、系统调用接口和操作系统之间的关系如图所示，它表现了操作系统如何处理一个调用open()系统调用的用户应用。



# 考点三：操作系统运行环境

## 系统调用过程

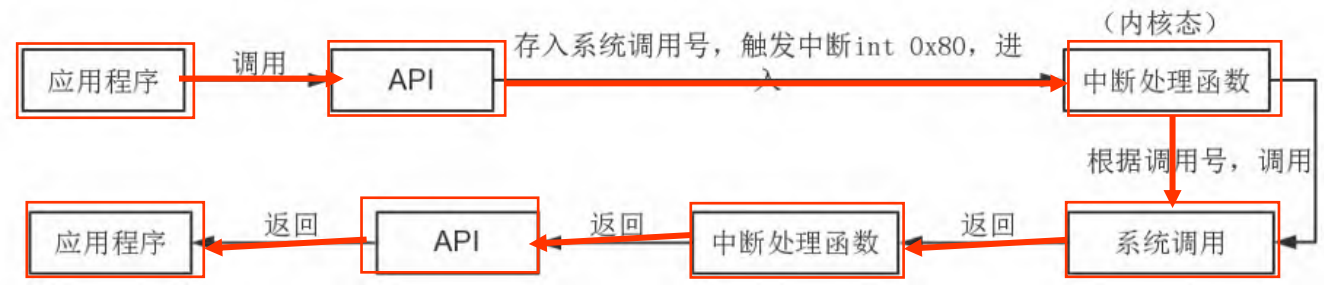



① 应用程序通过调用API，发起系统调用（请注意，此时程序处于用户态）。

② 当系统调用发生时，处理器通过一种特殊的机制（此处的中断编号是int 0x80），通常是中断或者异常处理，把控制流程转移到内核态的一些特定的位置（（请注意，通过中断，已经从用户态切换到了内核态）。处理器模式转变成特权模式。

# 考点三：操作系统运行环境

## 系统调用过程



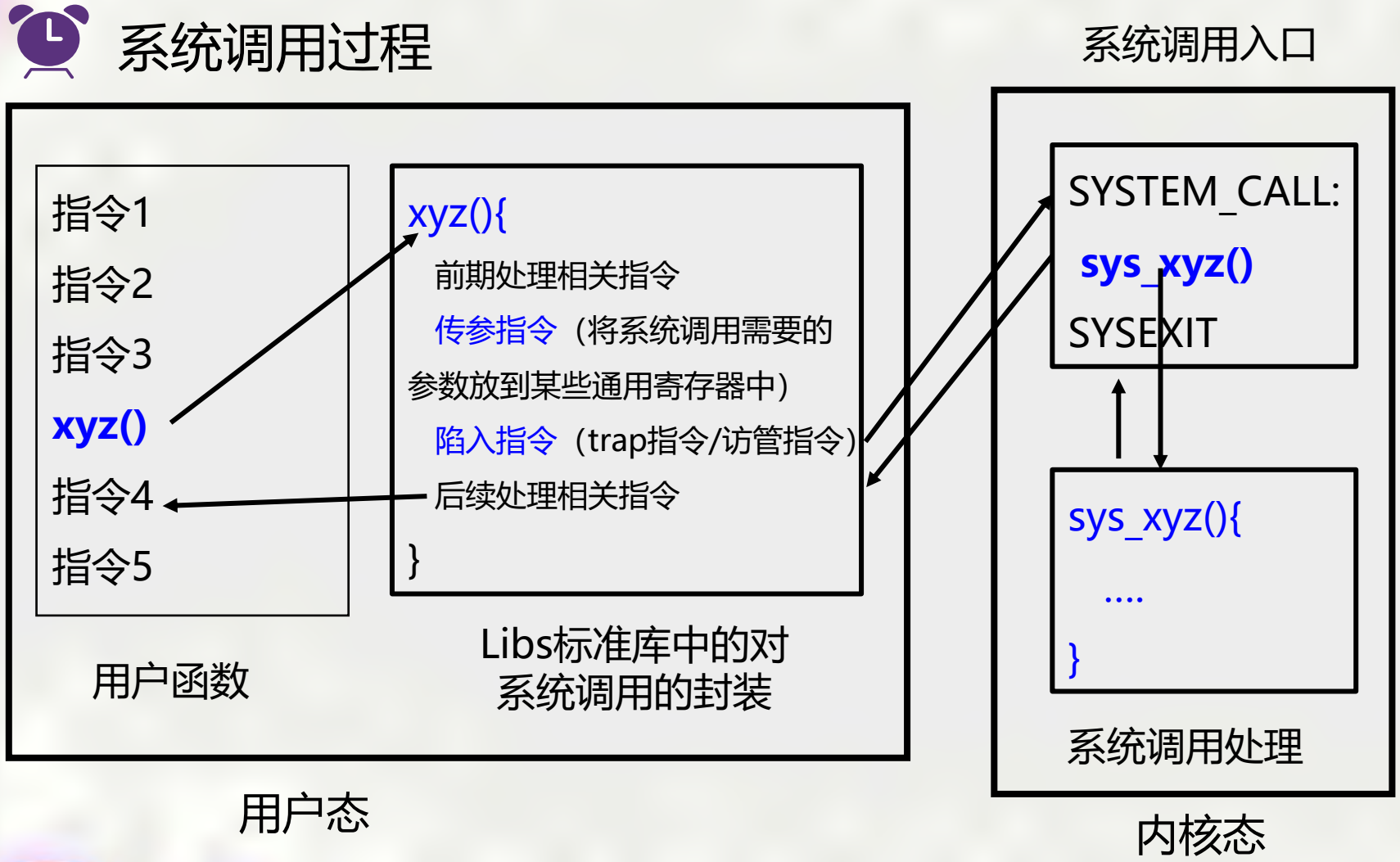

④ 由内核程序执行被请求的功能代码。这个功能代码代表着对一段标准程序段的执行，用以完成所请求的功能。

⑤ 处理结束之后，程序状态恢复系统调用之前的现场；把运行模式从特权模式恢复成为用户模式。

⑥ 最后通过API将控制权转移回原来的用户程序。

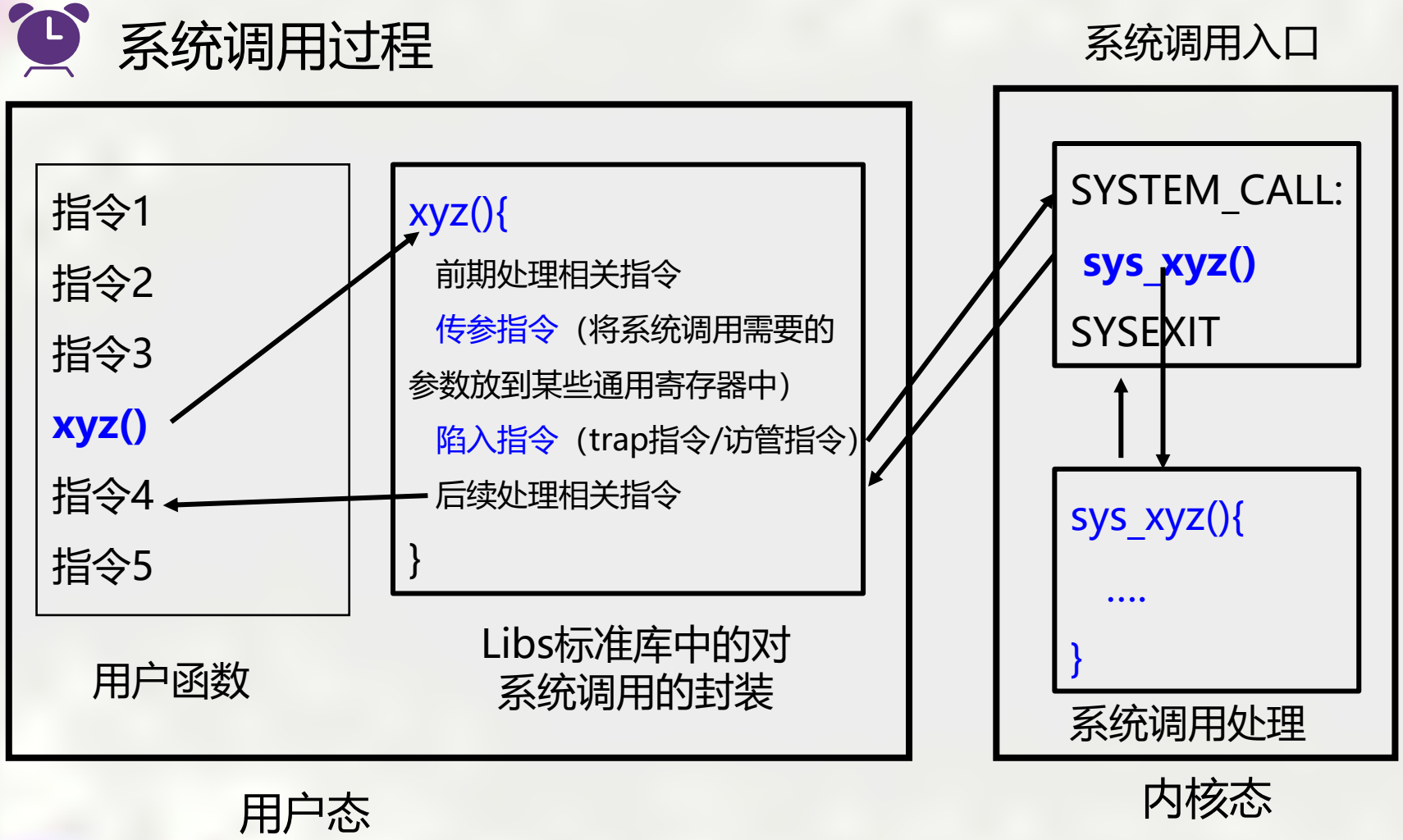
# 考点三：操作系统运行环境

## 系统调用过程



# 考点三：操作系统运行环境

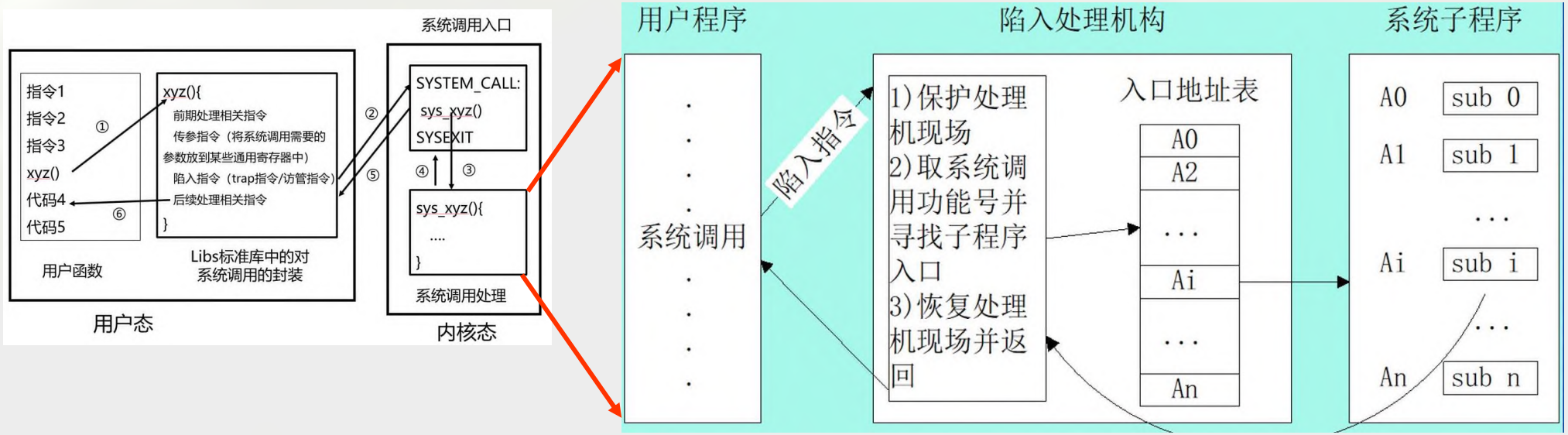
## 系统调用过程




传递系统调用参数 -> 执行陷入指令（用户态） -> 执行相应的内请求核程序处理系统调用（核心态） -> 返回应用程序

# 考点三：操作系统运行环境

## 系统调用过程




系统调用的执行过程如下：

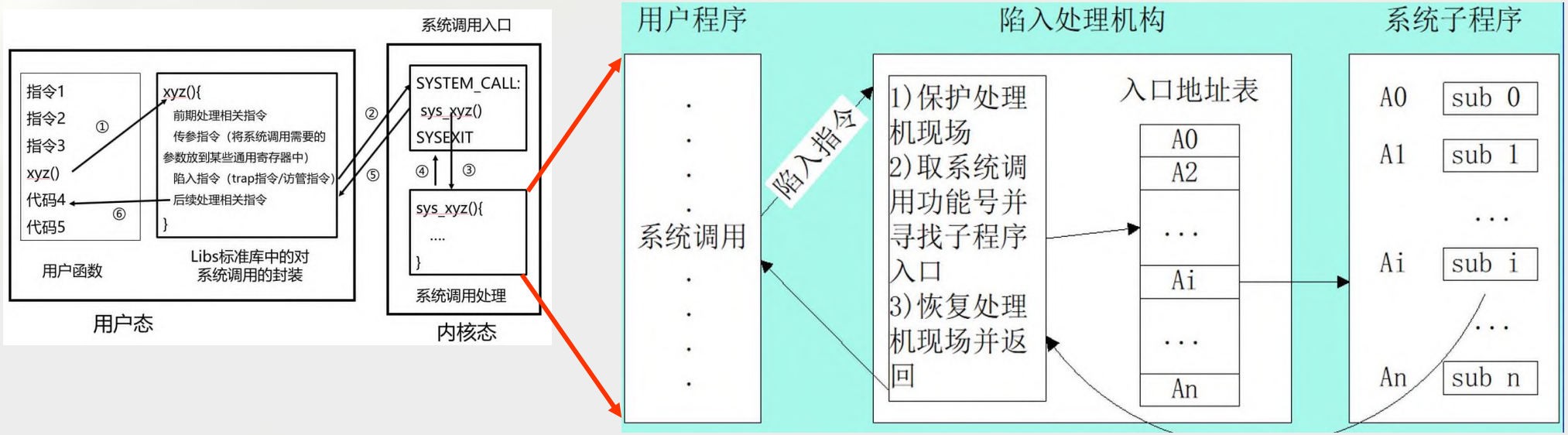
（1）应用程序 在 用户态 准备好调用参数，执行 int 指令触发 软中断，中断号为 0x80；

（2）CPU 被软中断打断后，执行对应的 中断处理函数，这时便已进入内核态；

（3）系统调用处理函数进入内核执行栈，并保存所有寄存器 (一般用汇编语言实现)；

# 考点三：操作系统运行环境

## 系统调用过程




系统调用的执行过程如下：

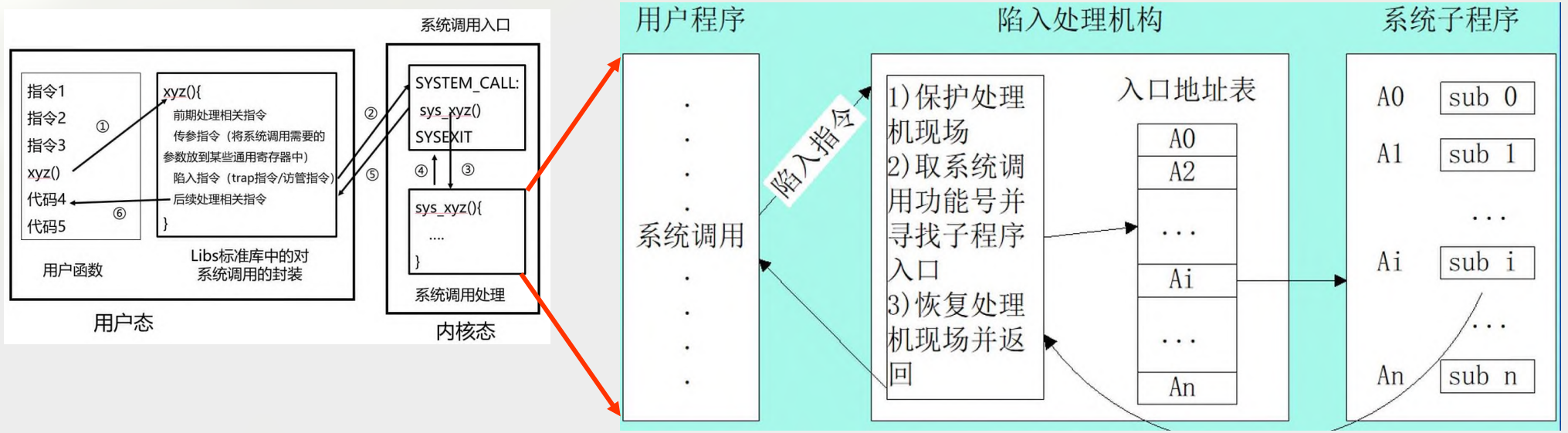
（4）系统调用处理函数根据系统调用号调用对应的 C 函数—— 系统调用服务例程；

（5）系统调用处理函数准备返回值并从内核栈中恢复寄存器；

（6）系统调用处理函数 执行 return 指令切换回用户态 。

# 考点三：操作系统运行环境

## 系统调用过程




请注意：（1）陷入指令是在用户态执行的，执行陷入指令之后立即引发一个内中断，使CPU进入核心态。

（2）发出系统调用请求是在用户态，而对系统调用的相应处理在核心态下进行。

# 考点三：操作系统运行环境

## 系统调用过程

| | 发生位置 | 处理位置 |
|---|---|---|
| **中断** | 用户态/内核态 | 内核态 |
| **异常** | 用户态/内核态 | 内核态 |
| **缺页** | 用户态/内核态 | 内核态 |
| **系统调用** | 用户态 | 内核态 |
| **进程创建** | 用户态/内核态 | 内核态 |
| **进程切换** | 内核态 | 内核态 |
| **进程调度** | 内核态 | 内核态 |

常见是事件的发生和处理位置

# 考点三：操作系统运行环境

## 系统调用类型

- 采用系统调用有诸多好处，应用程序通过系统调用请求操作系统的服务。
- 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；

# 考点三：操作系统运行环境

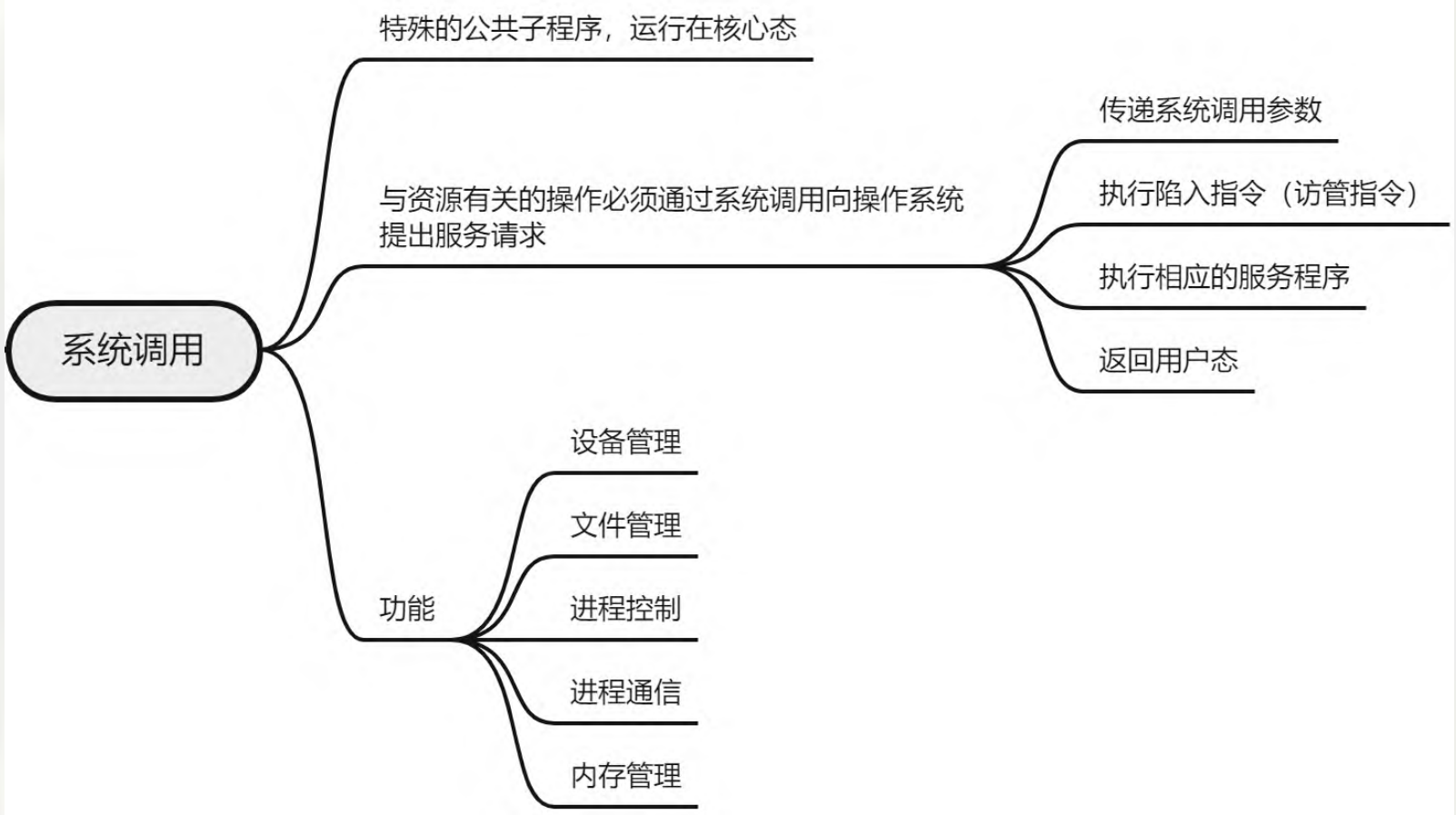
## 系统调用类型

| | |
|---|---|
| **进程控制** | 进程结束或者放弃 |
| | 进程装入和执行 |
| | 进程创建 |
| | 进程终止 |
| | 获取或者设置进程属性 |
| | 等待事假，唤醒时间 |
| | 分配和释放内存 |
| **文件管理** | 创建文件 |
| | 删除文件 |
| | 打开文件 |
| | 关闭文件 |
| | 读、写、重定位文件 |
| | 设置或者获取文件属性 |
| **设备管理** | 请求设备，释放设备 |
| | 读、写、重定位 |
| | 取得设备属性，设置设备属性 |
| | 逻辑连接或断开设备信息维护 |
| **进程通信** | 创建，删除通信连接 |
| | 发送，接受消息 |
| | 传递状态消息 |
| | 连接或断开远程设备 |

- 系统调用获得是操作系统提供的通用服务。
- 每一个系统调用都有独一无二的系统调用号
- 应用程序通过系统调用号调用。

# 考点三：操作系统运行环境

## 系统调用小结



【政哥点拨】

1．下列关于系统调用的说法，正确的是(　)。

Ⅰ．用户程序设计时，使用系统调用命令，该命令经过编译后，形成若干参数和陷入指令

Ⅱ．用户程序设计时，使用系统调用命令，该命令经过编译后，形成若干参数和屏蔽中断指令

Ⅲ．系统调用功能是操作系统向用户程序提供的接口

Ⅳ．用户及其应用程序和应用系统是通过系统调用提供的支持和服务来使用系统资源完成其操作的

A．仅Ⅰ、Ⅲ　　B．仅Ⅱ、Ⅳ　　C．Ⅰ、Ⅲ、Ⅳ　　D．Ⅱ、Ⅲ、Ⅳ

C【**解析**】Ⅰ正确：系统调用需要触发trap指令，如基于x86的Linux系统，该指令为int 0x80。

Ⅲ正确：系统调用功能是操作系统向用户程序提供的接口。

1．下列关于系统调用的说法，正确的是(　)。

Ⅰ．用户程序设计时，使用系统调用命令，该命令经过编译后，形成若干参数和陷入指令

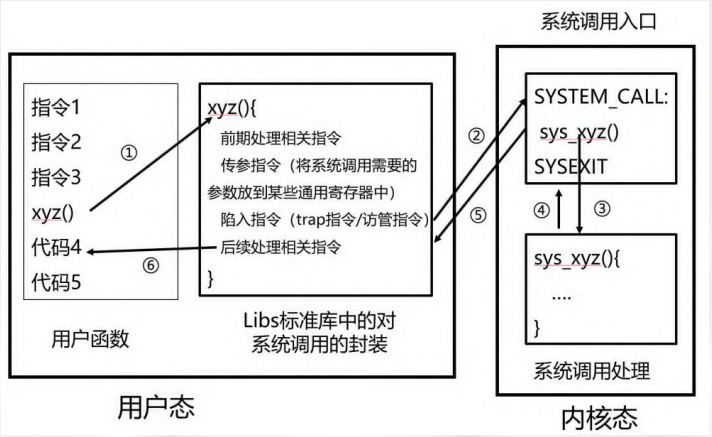
Ⅱ．用户程序设计时，使用系统调用命令，该命令经过编译后，形成若干参数和屏蔽中断指令

Ⅲ．系统调用功能是操作系统向用户程序提供的接口

Ⅳ．用户及其应用程序和应用系统是通过系统调用提供的支持和服务来使用系统资源完成其操作的

A．仅Ⅰ、Ⅲ　　B．仅Ⅱ、Ⅳ　　C．Ⅰ、Ⅲ、Ⅳ　　D．Ⅱ、Ⅲ、Ⅳ

C【**解析**】Ⅱ是干扰项，程序设计无法形成屏蔽中断指令。



1．下列关于系统调用的说法，正确的是(　)。

Ⅰ．用户程序设计时，使用系统调用命令，该命令经过编译后，形成若干参数和陷入指令

Ⅱ．用户程序设计时，使用系统调用命令，该命令经过编译后，形成若干参数和屏蔽中断指令

Ⅲ．系统调用功能是操作系统向用户程序提供的接口

Ⅳ．用户及其应用程序和应用系统是通过系统调用提供的支持和服务来使用系统资源完成其操作的

A．仅Ⅰ、Ⅲ　　B．仅Ⅱ、Ⅳ　　C．Ⅰ、Ⅲ、Ⅳ　　D．Ⅱ、Ⅲ、Ⅳ

C【**解析**】Ⅳ正确：操作系统是一层接口，对上层提供服务，对下层进行抽象。它通过系统调用向其上层的用户、应用程序和应用系统提供对系统资源的使用。

2．用户程序在用户态下要使用特权指令引起的中断属于(　)。

A．硬件故障中断　　B．程序中断

C．外部中断　　D．访管中断

D【**解析**】因操作系统不允许用户直接执行某些“危险性高”的指令，因此用户态运行这些指令的结果会转成操作系统的核心态去运行。这个过程就是访管中断。

访管指令是一条可以在用户态下执行的指令。当源程序中有需要操作系统服务的要求时，编译程序就会在由源程序转换成的目标程序中安排一条访管指令并设置一些参数。当目标程序执行时，中央处理器若取到了访管指令就产生一个中断事件，中断装置就会把中央处理器转换成管态，并让操作系统处理该中断事件。操作系统分析访管指令中的参数，然后让相应的“系统调用”子程序为用户服务。系统调用功能完成后，操作系统把中央处理器的管态改为用户态，并返回到用户程序。

3．处理器执行的指令被分为两类，其中有一类称为**特权指令**，它只允许(　)使用。

A．操作员　　B．联机用户

C．目标程序　　D．操作系统

D【**解析**】内核可以执行处理器能执行的任何指令，用户程序只能执行除特权指令外的指令。所以特权指令只能由内核即操作系统使用。

4．下列指令中，只能在内核态执行的是(　)。

A．trap指令　　B．I/O指令

C．数据传送指令　D．设置断点指令

| | 发生位置 | 处理位置 |
|---|---|---|
| 中断 | 用户态/内核态 | 内核态 |
| 异常 | 用户态/内核态 | 内核态 |
| 缺页 | 用户态/内核态 | 内核态 |
| 系统调用 | 用户态 | 内核态 |
| 进程创建 | 用户态/内核态 | 内核态 |
| 进程切换 | 内核态 | 内核态 |
| 进程调度 | 内核态 | 内核态 |

4．下列指令中，只能在内核态执行的是(　)。

A．trap指令　　B．I/O指令

C．数据传送指令　D．设置断点指令

B【**解析**】在内核态下CPU可执行任何指令，在用户态下CPU只能执行非特权指令，而特权指令只能在内核态下执行。常见的特权指令有：有关对I/O设备操作的指令；有关访问程序状态的指令；存取特殊寄存器的指令；其他指令。A、C和D都是提供给用户的指令。

【牛刀小试】

1．【广东工业大学 2017】计算机系统中设置的访管指令，（ ）执行。

A．只能在目态　　B．只能在管态

C．既可在目态又可在管态　　D．在目态和管态下都不能

A【**解析**】访管指令仅在用户态下使用，执行访管指令将用户态转变为核心态。

因此选择A。

- 访管指令是一条可以在用户态下执行的指令。当源程序中有需要操作系统服务的要求时，编译程序就会在由源程序转换成的目标程序中安排一条访管指令并设置一些参数。当目标程序执行时，中央处理器若取到了访管指令就产生一个中断事件，中断装置就会把中央处理器转换成管态，并让操作系统处理该中断事件。操作系统分析访管指令中的参数，然后让相应的“系统调用”子程序为用户服务。系统调用功能完成后，操作系统把中央处理器的管态改为用户态，并返回到用户程序。

- 为什么要在程序中引入访管指令呢？这是因为用户程序只能在用户态下运行，如果用户程序想要完成用户态下无法完成的工作，该怎么办呢？解决这个问题要靠访管指令。访管指令本身不是特权指令，其基本功能是让程序拥有“自愿进管”的手段，从而引起访管中断。
- 当处于用户态的用户程序使用访管指令时，系统根据访管指令的操作数执行访管中断处理程序，访管中断处理程序将按系统调用的操作数和参数转到相应的例行子程序。完成服务功能后，退出中断，返回到用户程序断点继续执行。

2．【南京工业大学 2015】下列指令中，不能在用户态执行的是(　)。

A．trap(访管)指令　　B．跳转指令

C．退栈指令　　D．关中断指令

| | 发生位置 | 处理位置 |
|---|---|---|
| 中断 | 用户态/内核态 | 内核态 |
| 异常 | 用户态/内核态 | 内核态 |
| 缺页 | 用户态/内核态 | 内核态 |
| 系统调用 | 用户态 | 内核态 |
| 进程创建 | 用户态/内核态 | 内核态 |
| 进程切换 | 内核态 | 内核态 |
| 进程调度 | 内核态 | 内核态 |

2．【南京工业大学 2015】下列指令中，不能在用户态执行的是(　)。

A．trap(访管)指令　　B．跳转指令

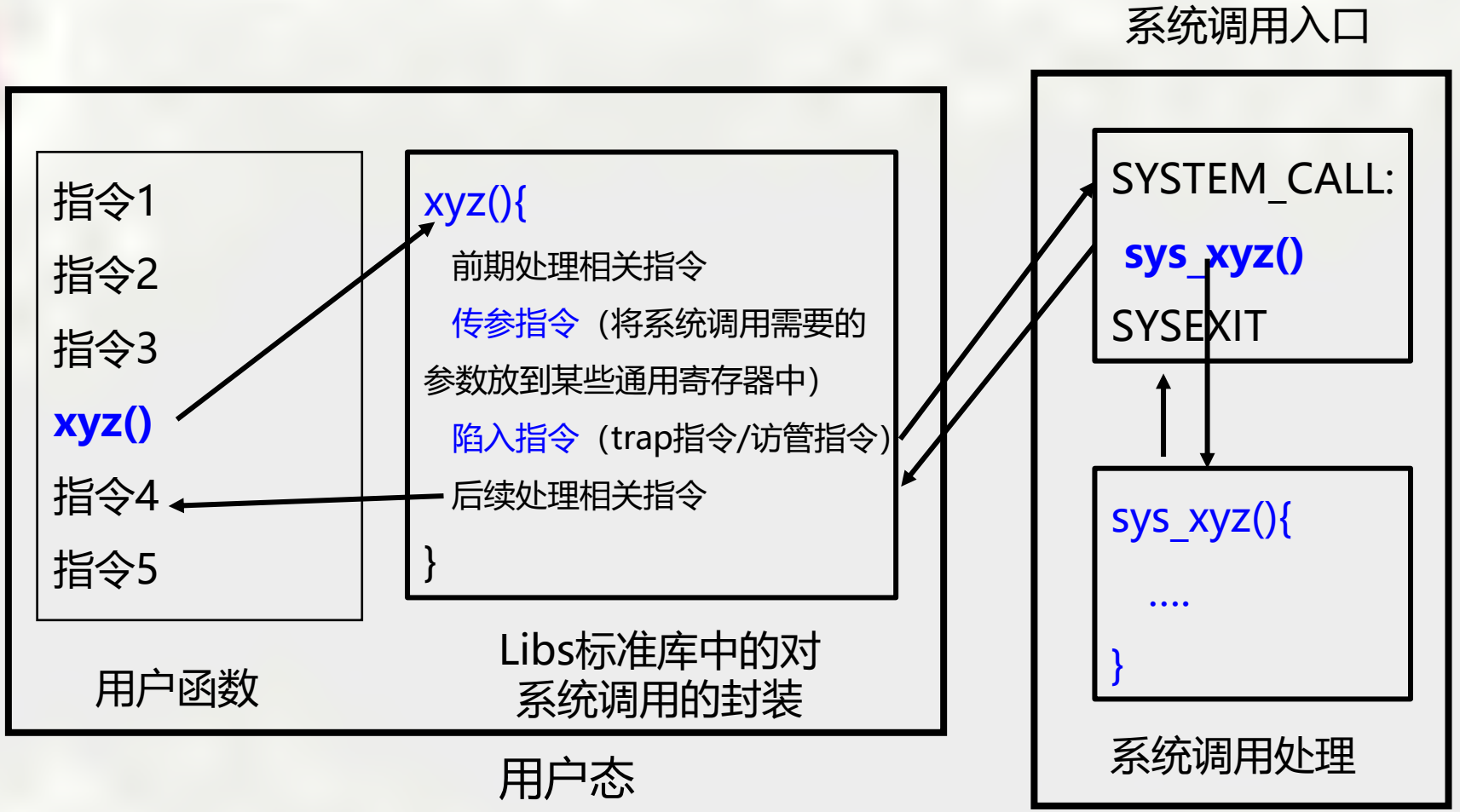
C．退栈指令　　D．关中断指令

D【**解析**】关中断指令必须在核心态才能执行，trap(访管)指令可以在用户态下执行，执行完就转到核心态，跳转与退栈指令都是可以在用户态下执行的指令。因此选择D。

3．【南京工业大学　2017】当用户程序执行访管指令时，中断装置将使中央处理器(　)工作。

A．维持在目态　　B．从目态转换到管态

C．维持在管态　　D．从管态转换到目态


系统调用入口
指令1
指令2
指令3
xyz()
指令4
指令5
xyz(){
前期处理相关指令
传参指令（将系统调用需要的参数放到某些通用寄存器中）
陷入指令（trap指令/访管指令）
后续处理相关指令
}
SYSTEM_CALL:
sys_xyz()
SYSEXIT
sys_xyz(){
....
}
用户函数
Libs标准库中的对系统调用的封装
用户态
系统调用处理
内核态

3．【南京工业大学　2017】当用户程序执行访管指令时，中断装置将使中央处理器(　)工作。

A．维持在目态　　B．从目态转换到管态

C．维持在管态　　D．从管态转换到目态

B【**解析**】当目标程序执行时，中央处理器若取到了访管指令就产生一个中断事件，中断装置就会把中央处理器转换成管态，并让操作系统处理该中断事件。操作系统分析访管指令中的参数，然后让相应的“系统调用”子程序为用户服务。系统调用功能完成后，操作系统把中央处理器的管态改为目态，并返回到用户程序。因此选择B。

4．【桂林电子科技大学　2018】从资源管理的角度出发，将处理器执行的指令分成两类，其中的特权指令只允许(　)使用。

A．应用程序　　B．联机用户

C．操作系统程序　　D．目标程序

- 特权指令：是指计算机中不允许用户直接使用的指令，如 I/O指令、置中断指令，存取用于内存保护的寄存器、送程序状态字到程序状态字寄存器等的指令（这几种记好，属于内核态），说明此时正在运行的是内核程序，此时可以执行特权指令。
- 非特权指令：“管理程序”(即用户自编程序）出于安全考虑不能执行的指令，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。应用程序只能使用“非特权指令”，如：加法指令、减法指令等。

4．【桂林电子科技大学　2018】从资源管理的角度出发，将处理器执行的指令分成两类，其中的特权指令只允许(　)使用。

A．应用程序　　B．联机用户

C．操作系统程序　　D．目标程序

C【解析】特权指令是在系统态时运行的指令，只允许操作系统使用，不允许应用程序使用。因此选择C。

5．【北京交通大学　2017】为方便系统调用处理，操作系统内部设立有系统调用表，以建立系统调用号和相应系统调用子程序入口地址之间的映射关系。系统调用表本质上是(　)。

A．函数指针数组　　B．中断向量表

C．逻辑设备表　　D．工作集

| | |
|---|---|
| **进程控制** | 进程结束或者放弃 |
| | 进程装入和执行 |
| | 进程创建 |
| | 进程终止 |
| | 获取或者设置进程属性 |
| | 等待事假，唤醒时间 |
| | 分配和释放内存 |
| **文件管理** | 创建文件 |
| | 删除文件 |
| | 打开文件 |
| | 关闭文件 |
| | 读、写、重定位文件 |
| | 设置或者获取文件属性 |

- 系统调用获得是操作系统提供的通用服务。
- 每一个系统调用都有独一无二的系统调用号
- 应用程序通过系统调用号调用。

5．【北京交通大学　2017】为方便系统调用处理，操作系统内部设立有系统调用表，以建立系统调用号和相应系统调用子程序入口地址之间的映射关系。系统调用表本质上是(　)。

A．函数指针数组　　B．中断向量表

C．逻辑设备表　　D．工作集

A【**解析**】系统调用表本质上就是一个数组，数组的每一个元素保存的是一个函数指针，函数名命名sys_系统调用函数名，这些代码由内核已经实现好了。因此选择A。

6．用户可以通过(　)两种方式来使用计算机。

A．命令接口和函数　　B．命令接口和系统调用

C．命令接口和文件管理　　D．设备管理方式和系统调用

6．用户可以通过(　)两种方式来使用计算机。

A．命令接口和函数　　B．命令接口和系统调用

C．命令接口和文件管理　　D．设备管理方式和系统调用

B【**解析**】用户可以通过以下两种方式来使用计算机：

- (1)命令方式，这是指由操作系统提供的一组联机命令（语言），用户可以通过键盘输入有关的命令来直接操作计算机系统；
- (2)系统调用方式，操作系统提供了一组系统调用，用户可在应用程序中通过调用相应的系统调用来操作计算机。因此选择B。

7．操作系统提供给编程人员的接口是(　)。

A．库函数　　B．高级语言

C．系统调用　　D．子程序

7．操作系统提供给编程人员的接口是(　)。

A．库函数　　B．高级语言

C．系统调用　　D．子程序

C【**解析**】人机接口指的是人控制计算机的接口，是应用程序编程接口(API)。API是一些预先定义的函数。目的是提供应用程序基于某软件或硬件得以访问一组操作系统服务的能力，而又无须访问源码，或理解内部工作机制的细节。操作系统提供给程序员的接口是系统调用，而库函数是通过系统调用所形成的特定功能的函数。因此选择C。

8．为了方便用户直接或间接地控制自己的作业，**操作系统向用户**提供了命令接口，该接口又可进一步分为(　)。

A．联机用户接口和脱机用户接口

B．程序接口和图形接口

C．联机用户接口和程序接口

D．脱机用户接口和图形接口

8．为了方便用户直接或间接地控制自己的作业，**操作系统向用户**提供了命令接口，该接口又可进一步分为(　)。

A．联机用户接口和脱机用户接口

B．程序接口和图形接口

C．联机用户接口和程序接口

D．脱机用户接口和图形接口

A【**解析**】程序接口、图形接口与命令接口三者并没有从属关系。按命令控制方式不同命令接口分为联机用户接口和脱机用户接口。因此选择A。

9．用户在程序中试图读某文件的第100个逻辑块，使用操作系统提供的(　　)接口。

A．系统调用　　B．键盘命令

C．原语　　D．图形用户

| | |
|---|---|
| **进程控制** | 进程结束或者放弃 |
| | 进程装入和执行 |
| | 进程创建 |
| | 进程终止 |
| | 获取或者设置进程属性 |
| | 等待事假，唤醒时间 |
| | 分配和释放内存 |
| **文件管理** | 创建文件 |
| | 删除文件 |
| | 打开文件 |
| | 关闭文件 |
| | 读、写、重定位文件 |
| | 设置或者获取文件属性 |

- 系统调用获得是操作系统提供的通用服务。
- 每一个系统调用都有独一无二的系统调用号
- 应用程序通过系统调用号调用。

9．用户在程序中试图读某文件的第100个逻辑块，使用操作系统提供的(　　)接口。

A．系统调用　　B．键盘命令

C．原语　　D．图形用户

A【**解析**】用户在程序中试图读某文件的第100个逻辑块，操作系统通过系统调用向用户程序提供服务。因此选择A。

10．操作系统与用户通信接口通常不包括(　)。

A．shell　　B．命令解释器

C．广义指令　　D．缓存管理指令

10．操作系统与用户通信接口通常不包括(　)。

A．shell　　B．命令解释器

C．广义指令　　D．缓存管理指令

D【解析】广义指令就是系统调用；而命令解释器属于命令接口；shell指命令解释器，也属于命令接口。系统中的缓存全部由操作系统管理，对用户是透明的，操作系统不提供管理系统缓存的系统调用。因此选择D。

11．下面关于系统调用的描述中，正确的是(　)。

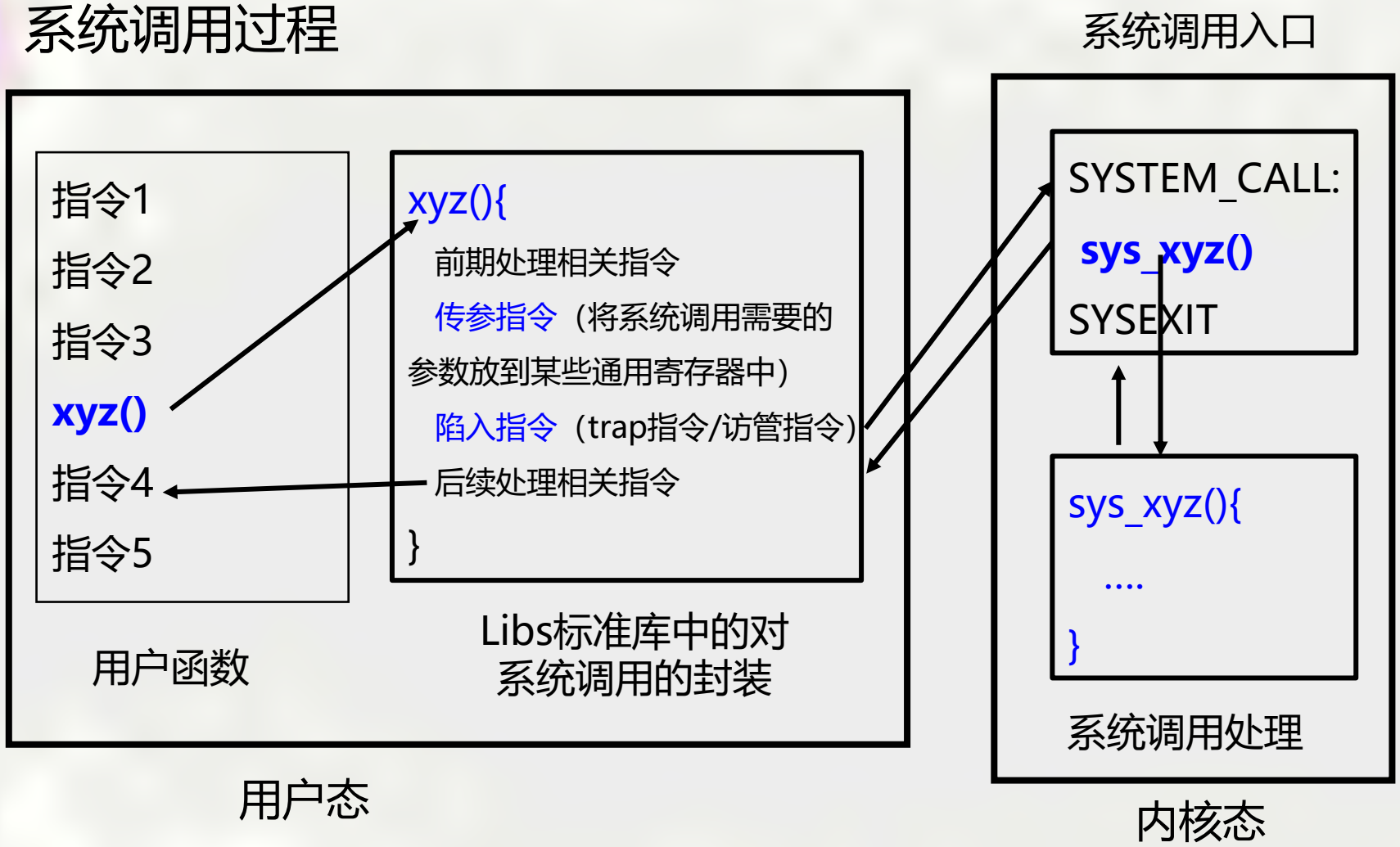
A．系统调用和中断都是程序主动发起的

B．当系统调用发生时，由用户进程独自处理

C．处理完系统调用后，操作系统会继续执行接下来的程序

D．处理系统调用时，执行的是系统内核函数

# 系统调用过程


系统调用入口
指令1
指令2
指令3
xyz()
指令4
指令5
xyz(){
前期处理相关指令
传参指令（将系统调用需要的参数放到某些通用寄存器中）
陷入指令（trap指令/访管指令）
后续处理相关指令
}
SYSTEM_CALL:
sys_xyz()
SYSEXIT
sys_xyz(){
....
}
用户函数
Libs标准库中的对系统调用的封装
系统调用处理
用户态
内核态

11．下面关于系统调用的描述中，正确的是(　)。

A．系统调用和中断都是程序主动发起的

B．当系统调用发生时，由用户进程独自处理

C．处理完系统调用后，操作系统会继续执行接下来的程序

D．处理系统调用时，执行的是系统内核函数

D【**解析**】系统调用是由应用程序发起的，是应用程序主动向操作系统发出服务请求，中断指的是程序被动停下来响应高优先级的代码再返回执行的过程；用户进程在执行系统调用时会陷入内核态，即存在内核态和用户态相互切换的情况；当处理器处理完系统调用操作后，操作系统会让处理器返回用户模式，继续执行用户代码；处理系统调用时，执行的是系统内核函数。因此选择D。

12．在操作系统中，只能在核心态下执行的指令是(　)。

A．读时钟　　B．寄存器清零

C．系统调用　　D．取数

| | 发生位置 | 处理位置 |
|---|---|---|
| 中断 | 用户态/内核态 | 内核态 |
| 异常 | 用户态/内核态 | 内核态 |
| 缺页 | 用户态/内核态 | 内核态 |
| 系统调用 | 用户态 | 内核态 |
| 进程创建 | 用户态/内核态 | 内核态 |
| 进程切换 | 内核态 | 内核态 |
| 进程调度 | 内核态 | 内核态 |

12．在操作系统中，只能在核心态下执行的指令是(　)。

A．读时钟　　B．寄存器清零

C．系统调用　　D．取数

C【**解析**】系统调用发生在用户态，执行在内核态，因此系统调用只能在核心态下执行。因此选择C。

- 系统调用发生在用户态，执行在内核态。外部中断也可以发生在用户态，执行在内核态。进程切换一定发生并且执行在内核态，这是操作系统的内核程序。

13．下列情况不会引起用户态和核心态切换的是(　　)。

A．进程执行过程中发生缺页

B．软件安装过程中需要用户输入序列号

C．一个进程需要调用另一个进程

D．打印过程中打印机出现故障

# 考点三：操作系统运行环境

用户态向内核态的转换

- **系统调用**：这是用户态进程主动要求切换到内核态的一种方式，用户态进程通过系统调用申请使用操作系统提供的服务程序完成工作，而系统调用的机制其核心还是使用了操作系统为用户特别开放的一个中断来实现。
- **异常**：当CPU在执行运行在用户态下的程序时，发生了某些事先不可知的异常，这时会触发由当前运行进程切换到处理此异常的内核相关程序中，也就转到了内核态，比如缺页异常。
- **外围设备的中断**：当外围设备完成用户请求的操作后，会向CPU发出相应的中断信号，这时CPU会暂停执行下一条即将要执行的指令转而去执行与中断信号对应的处理程序，如果先前执行的指令是用户态下的程序，那么这个转换的过程自然也就发生了由用户态到内核态的切换。

13．下列情况不会引起用户态和核心态切换的是(　　)。

A．进程执行过程中发生缺页

B．软件安装过程中需要用户输入序列号

C．一个进程需要调用另一个进程

D．打印过程中打印机出现故障

B 【**解析**】系统调用、异常、外围设备的中断是系统在运行时由用户态转到内核态的最主要方式。其中系统调用可以认为是用户进程主动发起的，异常和外围设备中断则是被动的。因此选择B。

14．下列关于中断和异常的描述，正确的是(　　)。

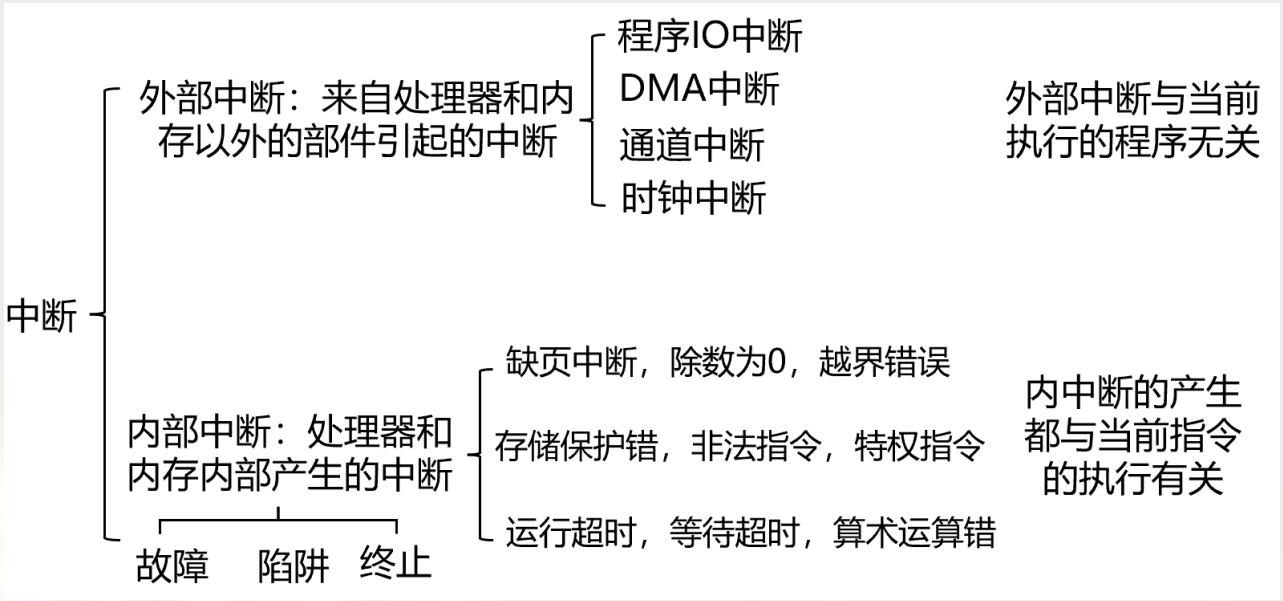
A．中断会产生异常

B．异常会产生中断

C．中断是异常的一种

D．中断和异常都与硬件故障有关

- 按照中断源的位置，可以分为内部中断和外部中断



14．下列关于中断和异常的描述，正确的是(　　)。

A．中断会产生异常

B．异常会产生中断

C．中断是异常的一种

D．中断和异常都与硬件故障有关

B【解析】异常会产生中断，导致程序的执行过程被打断。因此选择B。

15．操作系统提供了多种界面供用户使用，其中(　　)是专门供应用程序使用的一种界面。

A．终端命令　　B．图形用户窗口

C．系统调用　　D．作业控制语言


- 系统调用(system call)提供了操作系统提供的有效服务界面。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

15．操作系统提供了多种界面供用户使用，其中(　　)是专门供应用程序使用的一种界面。

A．终端命令　　B．图形用户窗口

C．系统调用　　D．作业控制语言

C【解析】操作系统提供了多种界面供用户使用，其中系统调用是专门供应用程序使用的一种界面。因此选择C。

16．下列关于库函数和系统调用的描述，正确的是(　)。

Ⅰ．库函数可以运行在用户空间

Ⅱ．有的库函数会使用系统调用

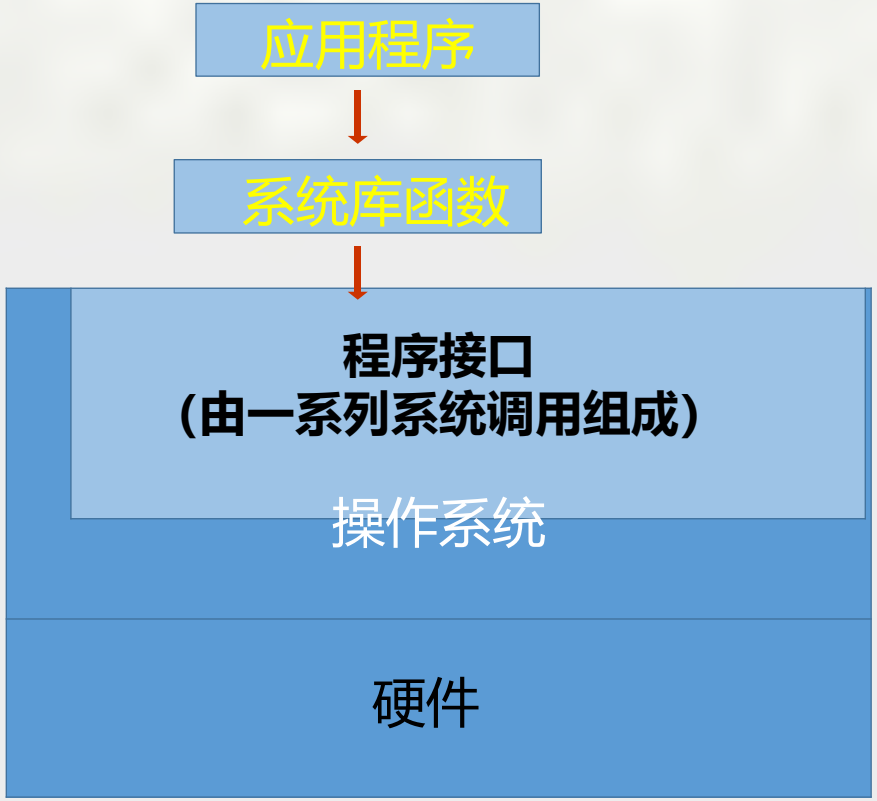
Ⅲ．系统调用的执行效率比库函数高

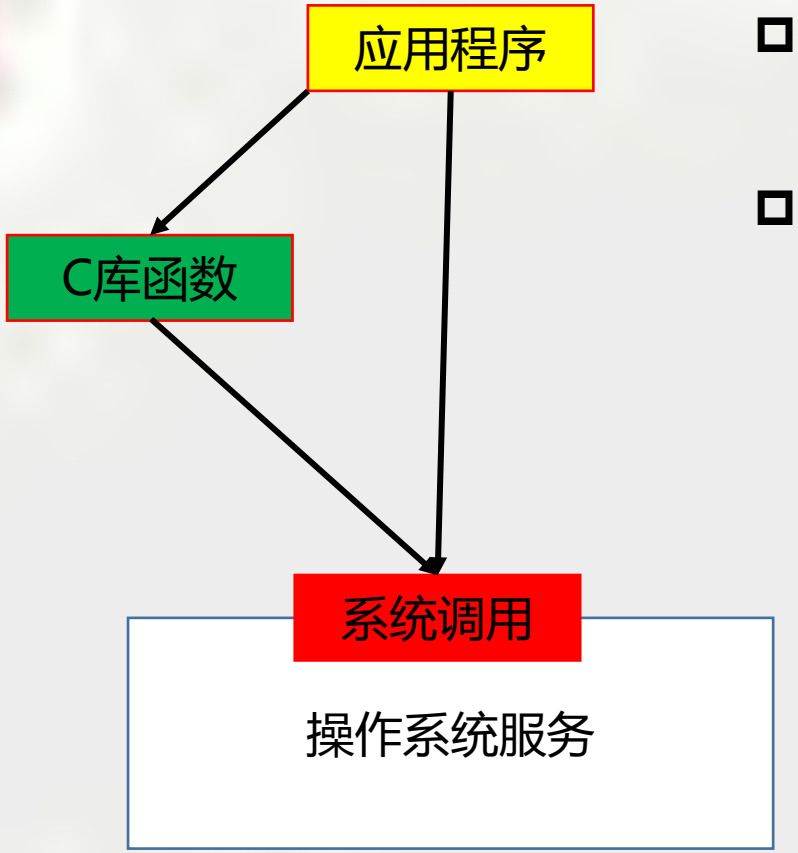
Ⅳ．当调用库函数时，必然会由用户态转为核心态

A．Ⅱ、Ⅲ、Ⅳ　　B．仅Ⅰ、Ⅲ

C．仅Ⅰ、Ⅱ　　D．Ⅰ、Ⅱ、Ⅲ

- 系统调用方式：OS提供了一组系统调用，用户可在自己的应用程序中通过相应的系统调用，来操纵计算机。






- 一般应用程序开发人员根据应用程序接口(API）设计程序。
- API是一系列适用于应用程序员的函数，包括传递给每个函数的参数及其返回的程序员想得到的值。

16．下列关于库函数和系统调用的描述，正确的是(　)。

Ⅰ．库函数可以运行在用户空间

Ⅱ．有的库函数会使用系统调用

Ⅲ．系统调用的执行效率比库函数高

Ⅳ．当调用库函数时，必然会由用户态转为核心态

A．Ⅱ、Ⅲ、Ⅳ　　B．仅Ⅰ、Ⅲ

C．仅Ⅰ、Ⅱ　　D．Ⅰ、Ⅱ、Ⅲ

C【**解析**】库函数是语言或应用程序的一部分，可以运行在用户空间中，而系统调用是操作系统的一部分，是内核为用户提供的程序接口，运行在内核空间中，而且许多库函数都会使用系统调用来实现功能。未使用系统调用的库函数，其执行效率通常要比使用系统调用的高，因为使用系统调用时，需要上下文的切换以及状态的切换（由用户态转为内核态）。因此选择C。

17．应该在**核心态**下运行的指令有(　)。

Ⅰ．屏蔽所有中断　　Ⅱ．读时钟周期

Ⅲ．设置时钟日期　　Ⅳ．存取某地址单元的内容

Ⅴ．停机

A．Ⅰ、Ⅲ、Ⅴ　　B．Ⅰ、Ⅱ、Ⅴ

C．Ⅱ、Ⅲ、Ⅳ　　D．仅Ⅰ、Ⅴ

| | **发生位置** | **处理位置** |
|---|---|---|
| **中断** | 用户态/内核态 | 内核态 |
| **异常** | 用户态/内核态 | 内核态 |
| **缺页** | 用户态/内核态 | 内核态 |
| **系统调用** | 用户态 | 内核态 |
| **进程创建** | 用户态/内核态 | 内核态 |
| **进程切换** | 内核态 | 内核态 |
| **进程调度** | 内核态 | 内核态 |

17．应该在**核心态**下运行的指令有(　)。

Ⅰ．屏蔽所有中断　　Ⅱ．读时钟周期

Ⅲ．设置时钟日期　　Ⅳ．存取某地址单元的内容

Ⅴ．停机

A．Ⅰ、Ⅲ、Ⅴ　　B．Ⅰ、Ⅱ、Ⅴ

C．Ⅱ、Ⅲ、Ⅳ　　D．仅Ⅰ、Ⅴ

A【解析】只在核心态下执行的指令有屏蔽所有中断、设置时钟日期、停机。因此选择A。

- 只在核心态执行的指令：屏蔽所有中断、设置时钟日期、停机。凡是涉及计算机根本运行的事情都应该在内核态下执行，而中断、时钟日期、存储映象图都属于系统级（相对应的是用户级）的资源，对这些资源的修改都必须在核心态，但是读取则没有强制要求。

18．下列关于操作系统的说法，错误的是(　　)。

A．只要计算机开机，就一直有程序运行

B．当电源接通时，计算机从核心态引导

C．不允许用户程序向包含中断向量的存储器单元写入数据

D．操作系统采用多道程序设计技术提高了CPU和外部设备的可靠性

D【**解析**】只要计算机开机，就一直有程序运行；当电源接通时，计算机从核心态引导；不允许用户程序向包含中断向量的存储器单元写入数据；操作系统采用多道程序设计技术，提高了CPU和外部设备的利用率。因此选择D。

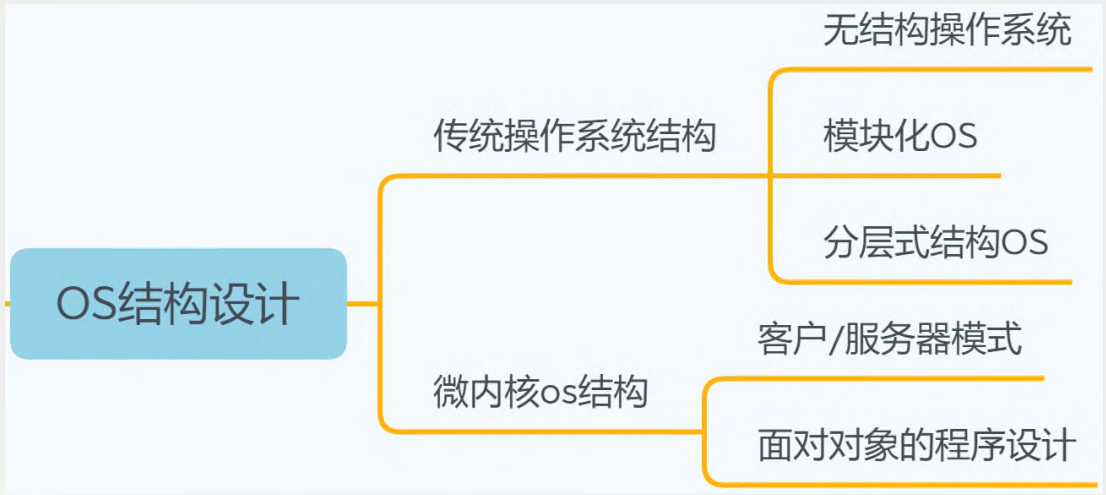
# 谢谢大家

# 考点四：操作系统的结构

# 考点框架

操作系统结构

微内核操作系统结构



# 考点四：操作系统的结构

## 操作系统结构

# 考点四：操作系统的结构

## 无结构操作系统

- 整体式结构也叫简单结构或无结构，在早期设计开发操作系统时，设计者只是把注意力放在功能的实现和获得高的效率上。
- 整个操作系统的功能由一个一个的过程来实现，这些过程之间又可以相互调用，导致操作系统变为一堆过程的集合，其内部结构复杂又混乱。因此这种操作系统没有结构可言。

# 考点四：操作系统的结构

## 无结构操作系统

- 优点就是接口简单直接，系统效率高
- 缺点：没有可读性，也不具备可维护性，一旦某一个过程出了问题，凡是与之存在调用关系的过程都要修改，所以给调试和维护人员带来许多麻烦，有时为了修改系统中的错误还不如重新设计开发一个操作系统。

# 考点四：操作系统的结构

## 模块化操作系统结构

- 模块化结构是指将整个操作系统按功能划分为若干个模块，每个模块实现一个特定的功能。
- 模块之间的通信只能通过预先定义的接口进行。或者说模块之间的相互关系仅限于接口参数的传递。

# 考点四：操作系统的结构

## 模块化操作系统结构

- 在模块化结构中，模块与模块之间的关联要尽可能地少，而模块内部的关联要尽可能地紧密，这样划分出来的模块之间具备一定的独立性，从而减少了模块之间的复杂的调用关系，使得操作系统的结构变得清晰：而模块内部各部分联系紧密，使得每个模块都具备独立的功能。

# 考点四：操作系统的结构

## 分层式操作系统结构

- 层次结构，就是把操作系统所有的功能模块按照功能调用次序分别排成若干层，各层之间的模块只有单向调用关系(例如，只允许上层或外层模块调用下层或内层模块)。

# 考点四：操作系统的结构

## 分层式操作系统结构

- 层次结构就和盖楼一样，最底层为硬件，最高层为用户层，每层只使用低层次的功能和服务，也就是说上层可以调用下层的服务但不可以调用其它层的服务

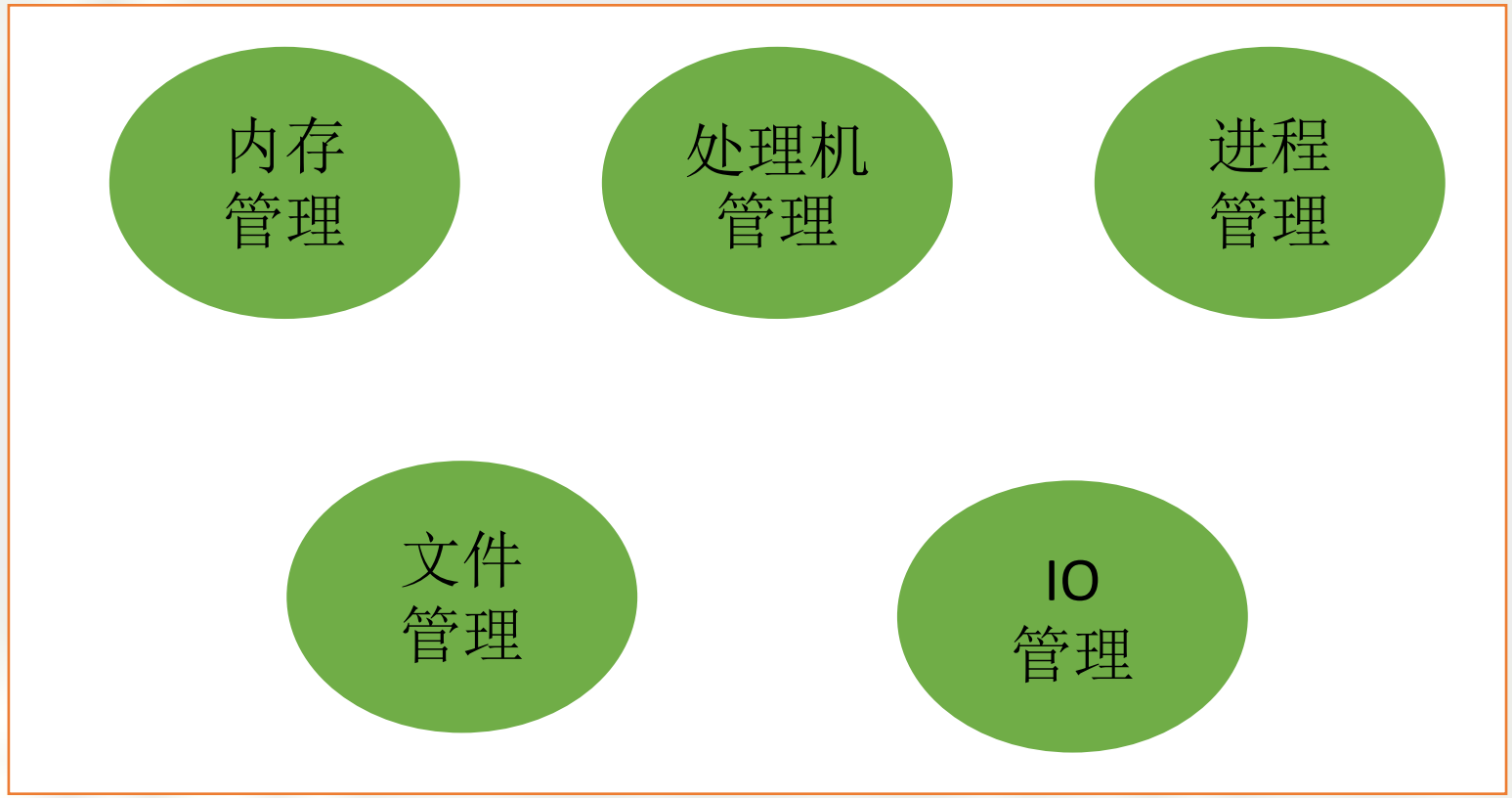
# 考点四：操作系统的结构

## 微内核操作系统结构

# 考点四：操作系统的结构

## 微内核操作系统结构

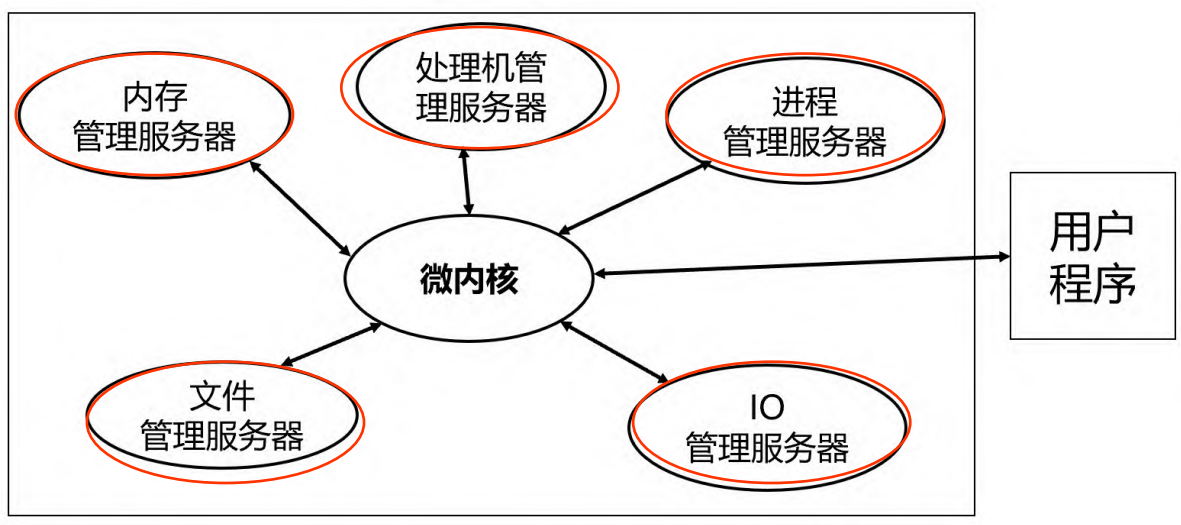
- 操作系统的功能



# 考点四：操作系统的结构

## 微内核操作系统结构

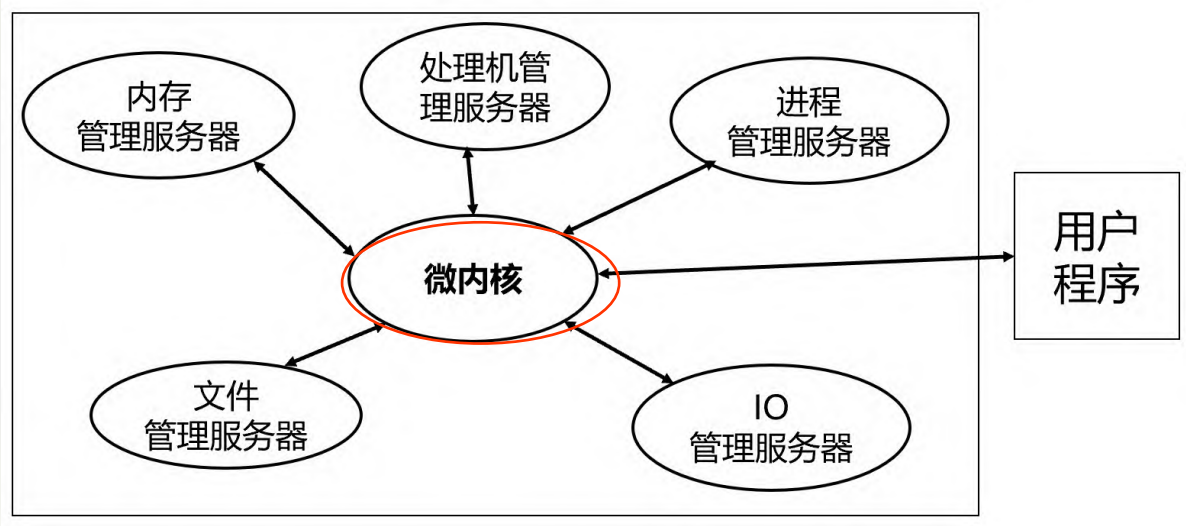
- 将操作系统划分为两部分: 一部分是用于提供各种服务的一组服务器(进程)，如用于提供进程管理的进程服务器、提供存储器管理的存储器服务器、提供文件管理的文件服务器等，所有这些服务器(进程)都运行在用户态。



# 考点四：操作系统的结构

## 微内核操作系统结构

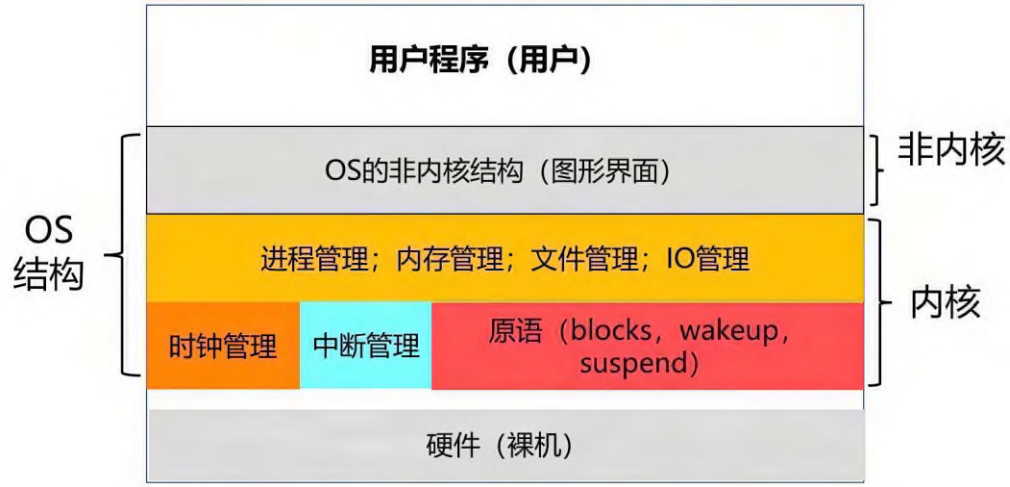
- 操作系统的另一部分是微内核，用来处理客户和服务器之间的通信，即由内核来接收客户的请求，再将该请求送至相应的服务器；同时它也接收服务器的应答，并将此应答回送给请求客户；



# 考点四：操作系统的结构

## 微内核操作系统结构

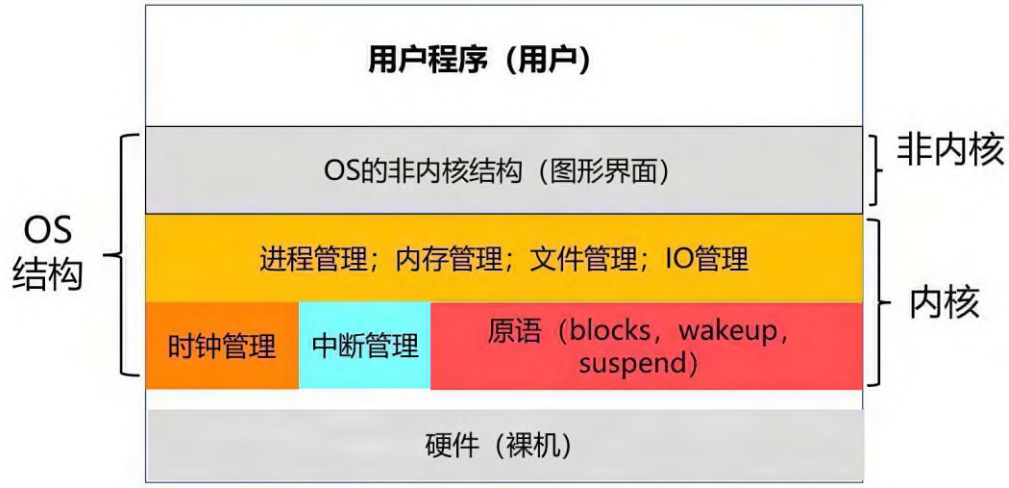
- 微内核是指精心设计的、能实现现代OS核心功能的小型内核，它与一般的OS(程序)不同， 它更小更精炼，它不仅运行在核心态，而且开机后常驻内存， 它不会因内存紧张而被换出内存。
- 微内核并非是一个完整的OS，而只是为构建通用OS提供一个重要基础。



# 考点四：操作系统的结构

## 微内核操作系统结构

- 微内核所提供的功能，通常都是一些最基本的功能且运行频率较高的程序，如进程管理、存储器管理、进程间通信、低级I/O功能。
- 在内核中还应具有其它一些机构，用于实现与硬件紧密相关的和一些较基本的功能，如时钟管理、中断处理、设备驱动等处于最低层。



# 考点四：操作系统的结构

## 微内核操作系统结构

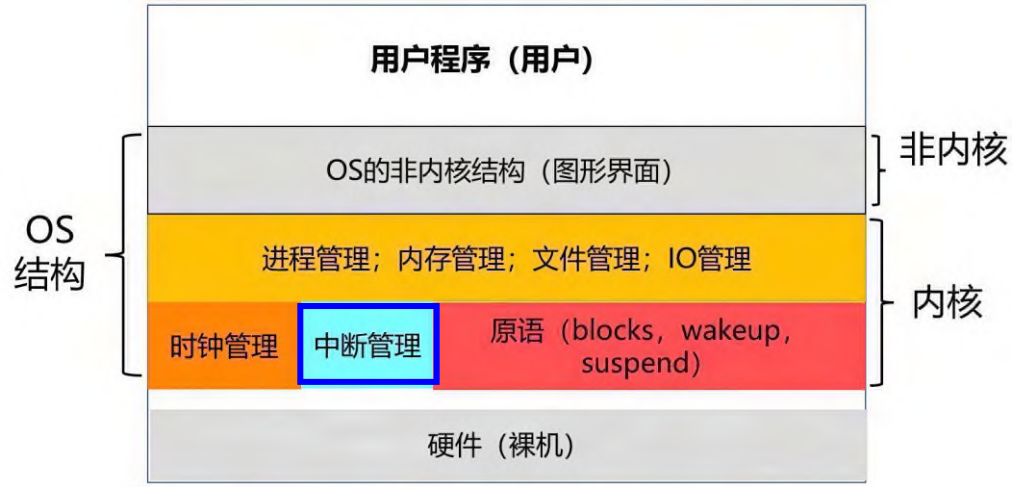
### 时钟管理

- 时钟的第一功能是计时，操作系统需要通过时钟管理，向用户提供标准的系统时间。
- 通过时钟中断的管理，可以实现进程的切换。如，在分时操作系统中采用时间片轮转调度，在实时系统中按截止时间控制运行。在批处理系统中通过时钟管理来衡量一个作业的运行程度等。

# 考点四：操作系统的结构

## 微内核操作系统结构

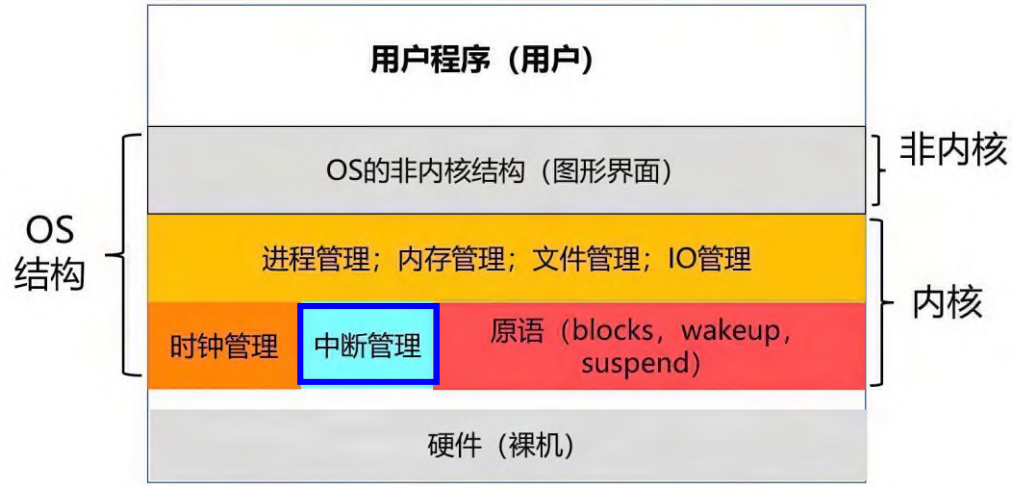
- 中断机制
- 引入中断技术提高多道程序运行环境中CPU的利用率，而且主要是针对外部设备的。键盘或鼠标信息的输入、进程的管理和调度、系统功能的调用、设备驱动、文件访问等，依赖于中断机制。现代操作系统是靠中断驱动的软件。



# 考点四：操作系统的结构

## 微内核操作系统结构

- 中断机制
- 中断机制中，只有一小部分功能属于内核，它们负责保护和恢复中断现场的信息，转移控制权到相关的处理程序。这样可以减少中断的处理时间，提高系统的并行处理能力。



# 考点四：操作系统的结构

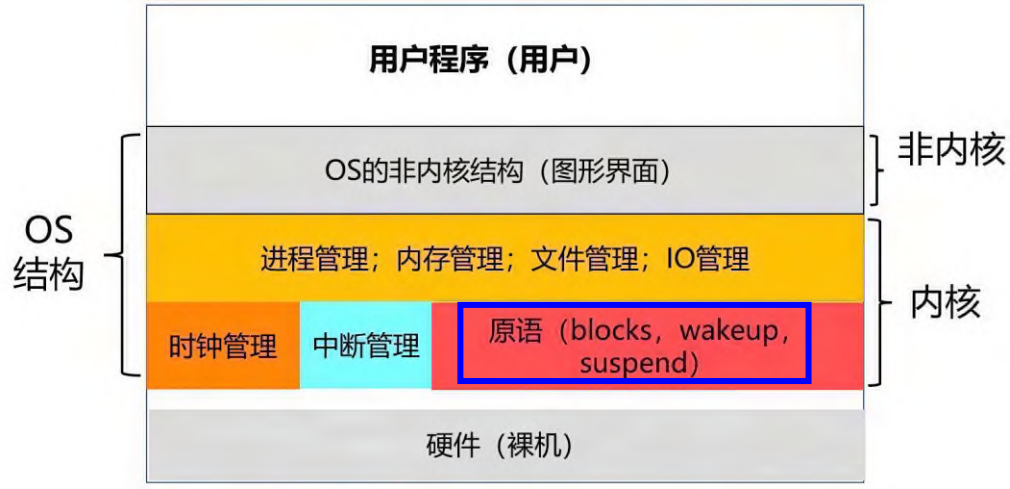
## 微内核操作系统结构

### 原语

按层次结构设计的操作系统，底层必然是一些可被调用的公用小程序，它们各自完成一个规定的操作。它们的特点如下:

①处于操作系统的最低层，是最接近硬件的部分。

②这些程序的运行具有原子性，其操作只能一气呵成



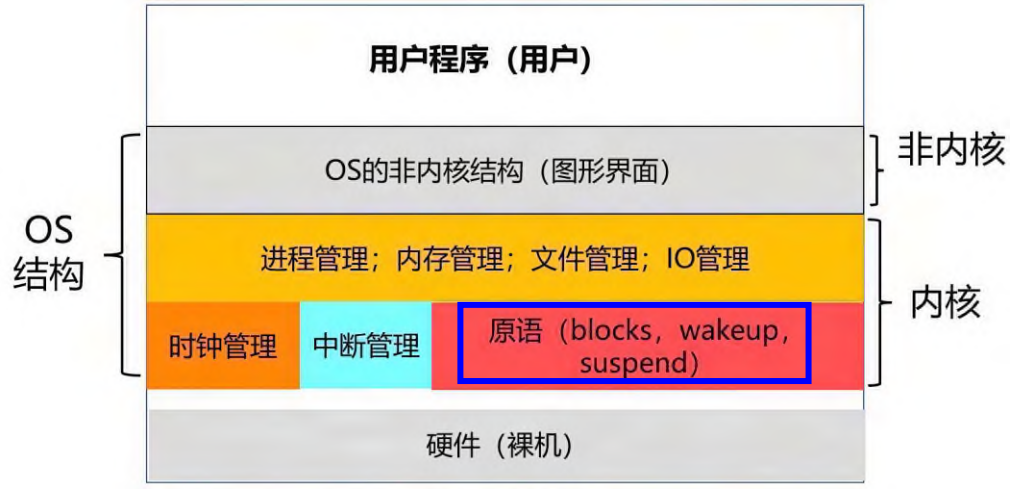
# 考点四：操作系统的结构

## 微内核操作系统结构

### 原语

按层次结构设计的操作系统，底层必然是一些可被调用的公用小程序，它们各自完成一个规定的操作。它们的特点如下:

③这些程序的运行时间都较短,而且调用频繁。



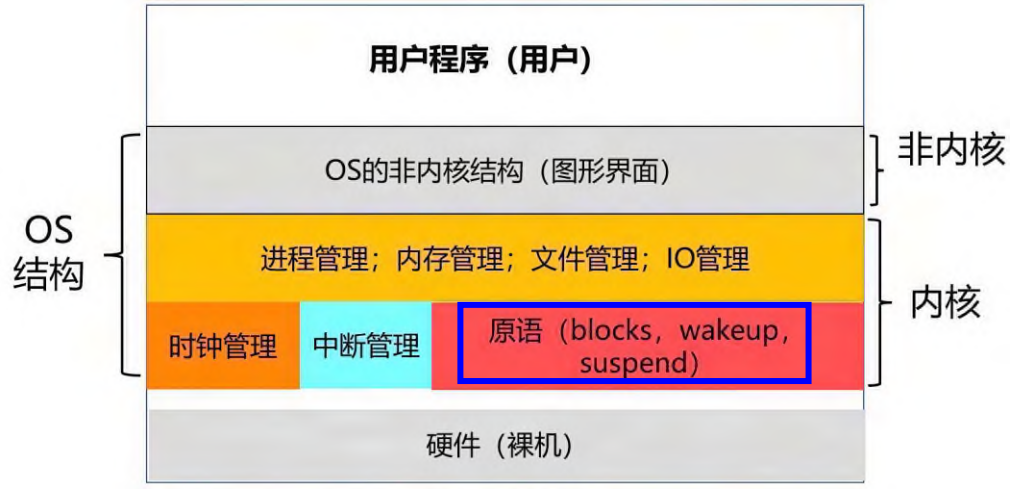
# 考点四：操作系统的结构

## 微内核操作系统结构

### 原语

按层次结构设计的操作系统，底层必然是一些可被调用的公用小程序，它们各自完成一个规定的操作。它们的特点如下:

④通常把具有这些特点的程序称为原语(Atomic Operation)。定义原语的直接方法是关闭中断，让其所有动作不可分割地完成后再打开中断。



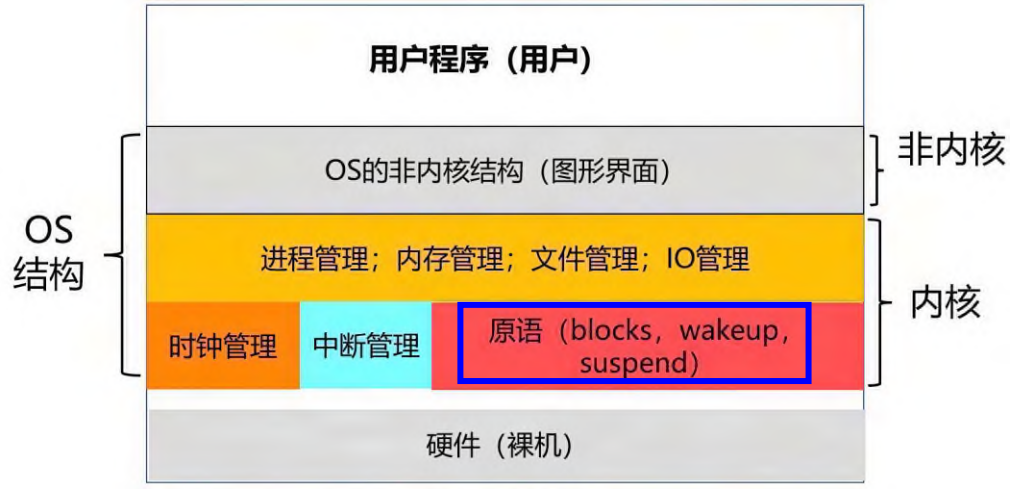
# 考点四：操作系统的结构

## 微内核操作系统结构

### 原语

按层次结构设计的操作系统，底层必然是一些可被调用的公用小程序，它们各自完成一个规定的操作。它们的特点如下:

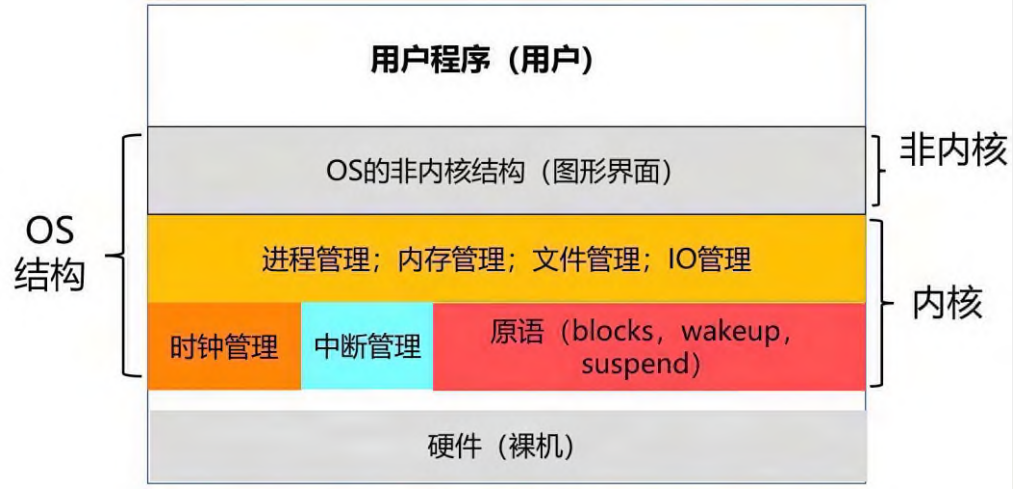
⑤系统中的设备驱动、CPU切换、进程通信等功能中的部分操作都可定义为原语，使它们成为内核的组成部分。



# 考点四：操作系统的结构

## 微内核操作系统结构

- 系统控制的数据结构及处理
  - 系统中用来登记状态信息的数据结构很多，如作业控制块、进程控制块（PCB)、设备控制块、各类链表、消息队列、缓冲区、空闲区登记表、内存分配表等。为了实现有效的管理，系统需要一些基本的操作，常见的操

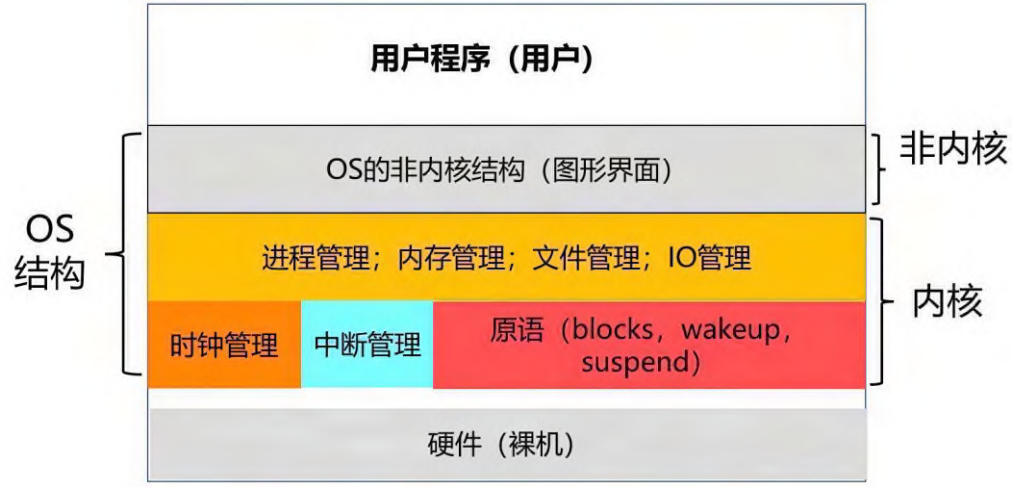


# 考点四：操作系统的结构

## 微内核操作系统结构

- 系统控制的数据结构及处理

- ①进程管理。进程状态管理、进程调度和分派、创建与撤销进程控制块等。
- ②存储器管理。存储器的空间分配和回收、内存信息保护程序、代码对换程序等。
- ③设备管理。缓冲区管理、设备分配和回收等。



# 考点四：操作系统的结构

## 微内核操作系统结构

内核的功能：（1）进程（线程）管理 （进程或者线程的调度）（2）低级存储器管理 （用户程序逻辑空间到内存空间的物理地址的变换）（3）中断和陷入管理 （中断和陷入）。具体来说，内核的功能包括：

（1）I/O指令、置终端屏蔽指令、清内存、建存储保护、设置时钟指令。

（2）中断、异常、陷入，比如缺页中断等

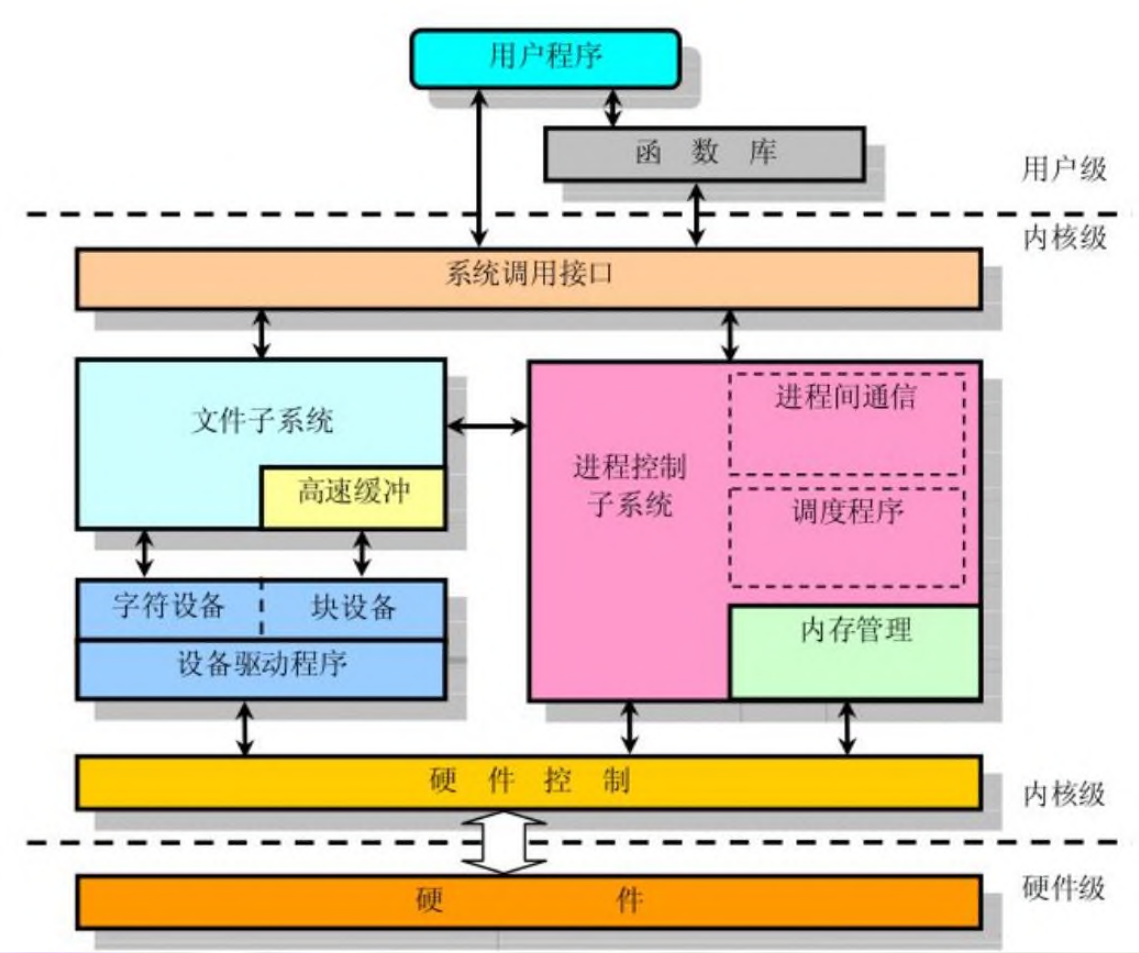
（3）进程（线程）管理

（4）系统调用，比如调用了设备驱动程序

（5）用户内存地址的转换（逻辑---> 物理映射）

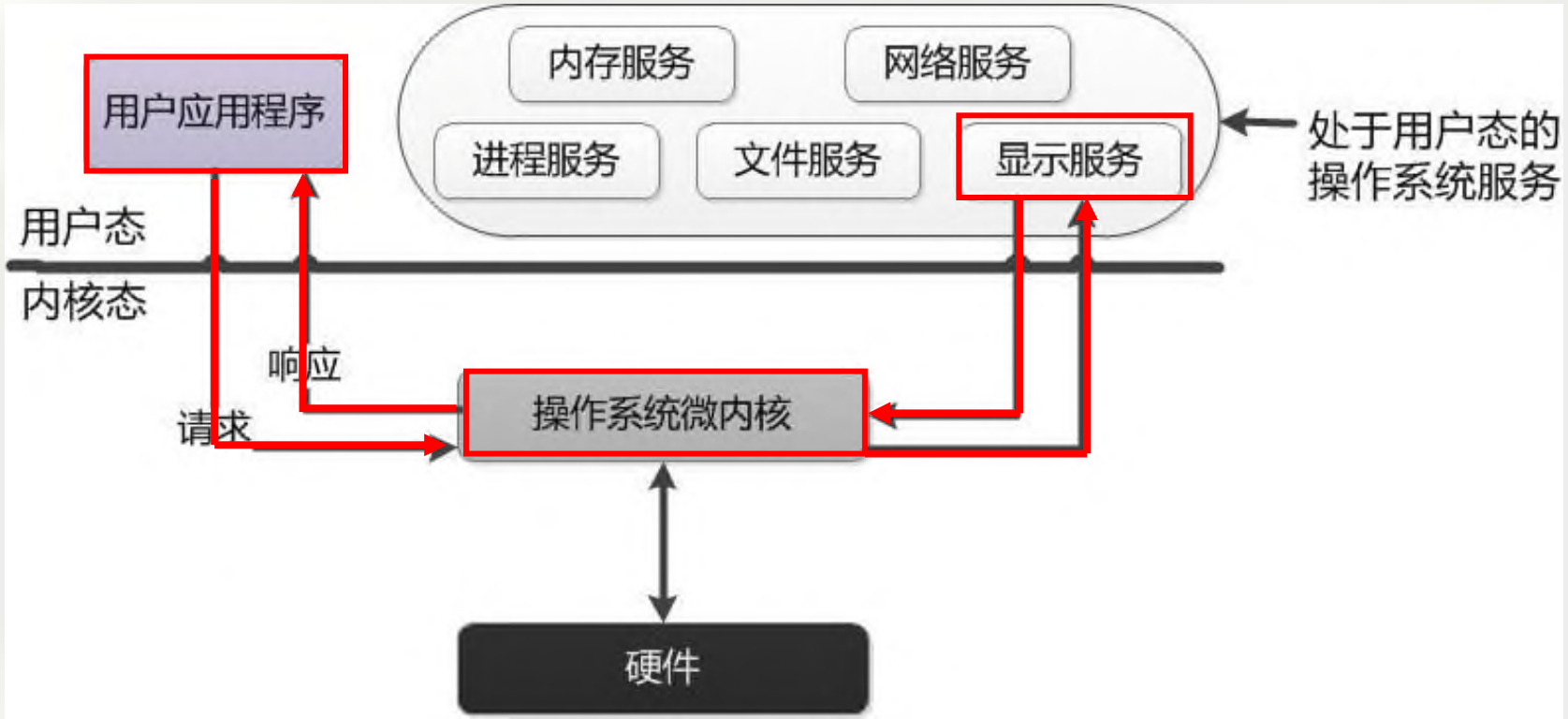
# 考点四：操作系统的结构

## 微内核操作系统结构



# 考点四：操作系统的结构

## 微内核操作系统结构



## 考点四：操作系统的结构

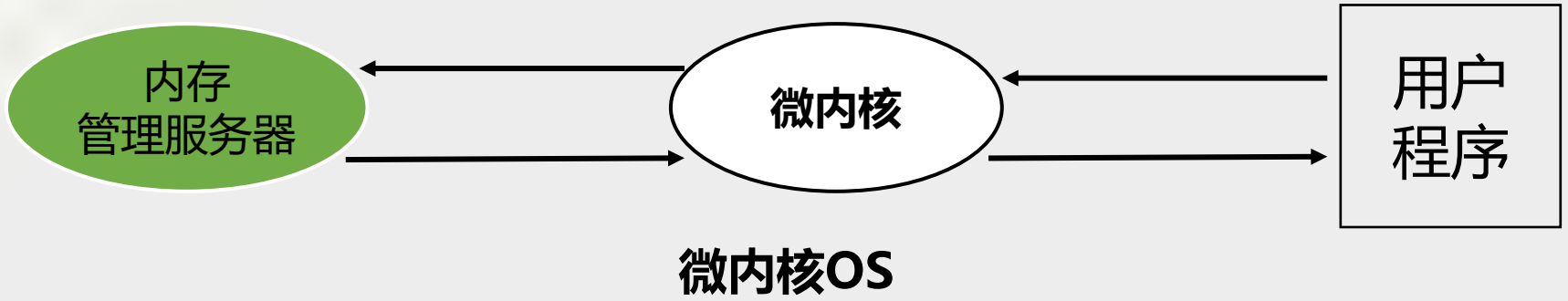
### 微内核操作系统结构

- 微内核将最基础的功能和与硬件有关的操作均放在内核中，其他各种服务器均与硬件无关，因此具有扩展性高，灵活性强；可靠性好和安全性高等优点；
- 由于微内核处于核心地位，也易于实现客户和服务器之间、服务器和服务器之间的通信采用消息传递机制。

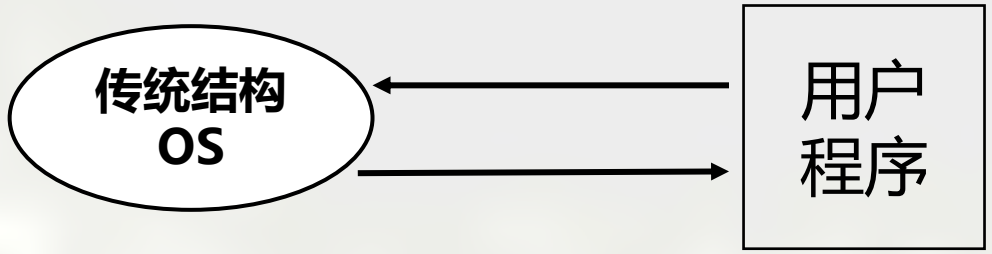
# 考点四：操作系统的结构

## 微内核操作系统结构

- 微内核好处多多，但是效率低




**微内核OS**




**传统结构OS**

【牛刀小试】

1．用(　)设计的操作系统结构清晰且便于调试。

A．分层式构架　　B．模块化构架

C．微内核构架　　D．宏内核构架

A【**解析**】操作系统结构的设计方法有无序模块法、微内核法、层次结构法及模块设计法等。各种设计方法总的目标都要保证操作系统工作的可靠性。

其中，层次结构法的最大特点是把整体问题局部化。一个大型复杂的操作系统被分解成若干单向依赖的层次，由各层的正确性来保证整个操作系统的正确性。采用层次结构不仅结构清晰，而且便于调试，有利于功能的增加、删减和修改。因此选择A。

2．下列关于分层式结构操作系统的说法，错误的是(　)。

A．各层之间只能是单向依赖或单向调用

B．容易实现在系统中增加或替换一层而不影响其他层

C．具有非常灵活的依赖关系

D．系统效率较低

C【解析】分层法的优点：①方便系统的调试和验证，简化了设计和实现，可以只是调试和验证一个层次而无须考虑其他层次。②易于扩展和维护，在系统中增加、修改或者替换一层中的模块或者整个层级的时候，只要不改变层间的接口就不影响其他层。

分层法的缺点：①合理定一个层比较困难，而且由于依赖关系的固化，往往会不太灵活。②效率低，操作系统每次执行一个功能都要穿过许多层，导致各层之间都需要通信机制，造成额外开销。

3．相对于微内核系统，(　)不属于大内核操作系统的缺点。

A．占用内存空间大

B．缺乏可扩展性而不方便移植

C．内核切换太慢

D．可靠性较低

C【**解析**】大内核：高性能、内核代码大、结构混乱、难以维护；微内核：内核功能少、结构清晰、方便管理、更加稳定、性能低。因此选择C。

4．下列说法中，(　)不适合描述微内核操作系统。

A．内核足够小　　B．功能分层设计

C．基于C/S模式　　D．策略与机制分离

B【**解析**】微内核操作系统：①足够小的内核；②基于客户/服务器模式；③应用“机制与策略分离”原理；④采用面向对象技术。因此选择B。

4．下列说法中，(　)不适合描述微内核操作系统。

A．内核足够小　　B．功能分层设计

C．基于C/S模式　　D．策略与机制分离

B【**解析**】微内核操作系统：①足够小的内核；②基于客户/服务器模式；③应用“机制与策略分离”原理；④采用面向对象技术。因此选择B。

微内核操作系统：(1)足够小的内核。在微内核操作系统中，内核是指精心设计的、能实现现代OS最基本的核心功能的部分。微内核并非是一个完整的OS，而只是操作系统中最基本的部分，它通常用于：

①实现与硬件紧密相关的处理；

②实现一些较基本的功能；

③负责客户和服务器之间的通信。它们只是为构建通用OS提供一个重要基础，这样就可以确保把操作系统内核做得很小。

微内核操作系统：(2)基于客户/服务器(Client/Server)模式。由于客户/服务器模式具有非常多的优点，故在单机微内核操作系统中几乎无一例外地都采用客户/服务器模式，将操作系统中最基本的部分放入内核中，而把操作系统的绝大部分功能都放在微内核外面的一组服务器（进程）中实现。例如，用于提供对进程（线程）进行管理的进程（线程）服务器、提供虚拟存储器管理功能的虚拟存储器服务器、提供I/O设备管理的I/O设备管理服务器等，它们都是被作为进程来实现的，运行在用户态，客户与服务器之间是借助微内核提供的消息传递机制来实现信息交互的。

微内核操作系统： (3)应用“机制与策略分离”原理。在现代操作系统的结构设计中，经常利用“机制与策略分离”的原理来构造OS结构。所谓机制，是指实现某一功能的具体执行机构。而策略，则是在机制的基础上，借助于某些参数和算法来实现该功能的优化，或达到不同的功能目标。通常，机制处于一个系统的基层，而策略则处于系统的高层。在传统的OS中，将机制放在OS的内核的较低层，把策略放在内核的较高层次中。而在微内核操作系统中，通常将机制放在OS的微内核中。正因为如此，才有可能将内核做得很小。

微内核操作系统： (4)采用面向对象技术。操作系统是一个极其复杂的大型系统软件，我们不仅可以通过结构设计来分解操作系统的复杂度，还可以基于面向对象技术中的“抽象”和“隐蔽”原则控制系统的复杂性，再进一步利用“对象”“封装”和“继承”等概念来确保操作系统的“正确性”“可靠性”“易修改性”“易扩展性”等，并提高操作系统的设计速度。正因为面向对象技术能带来如此多的好处，所以面向对象技术被广泛应用于现代操作系统的设计中。

5．对于以下五种服务，在采用微内核结构的操作系统中，（ )不宜放在微内核中。

Ⅰ．进程间通信机制　　Ⅱ．低级I/O

Ⅲ．低级进程管理和调度　Ⅳ．中断和陷入处理　　Ⅴ．文件系统服务

A．Ⅰ、Ⅱ和Ⅲ　　B．Ⅱ和Ⅴ

C．仅Ⅴ　　D．Ⅳ和Ⅴ

内核的功能包括：

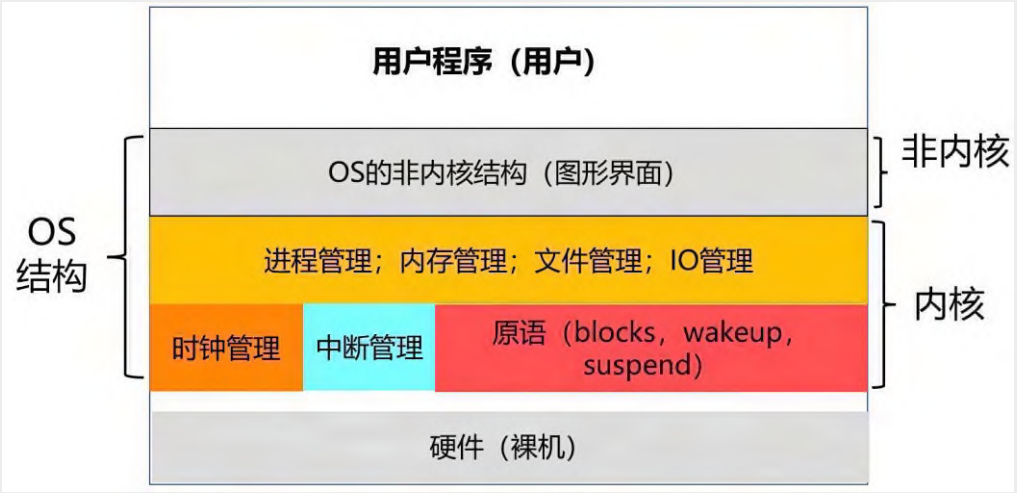
（1）I/O指令、置终端屏蔽指令、清内存、建存储保护、设置时钟指令。

（2）中断、异常、陷入，比如缺页中断等

（3）进程（线程）管理

（4）系统调用，比如调用了设备驱动程序

（5）用户内存地址的转换（逻辑---> 物理映射）



5．对于以下五种服务，在采用微内核结构的操作系统中，（ )不宜放在微内核中。

Ⅰ．进程间通信机制　　Ⅱ．低级I/O

Ⅲ．低级进程管理和调度　Ⅳ．中断和陷入处理　　Ⅴ．文件系统服务

A．Ⅰ、Ⅱ和Ⅲ　　B．Ⅱ和Ⅴ

C．仅Ⅴ　　D．Ⅳ和Ⅴ

C【解析】微内核几乎不做任何工作，提供以下4种服务：①进程间通信机制；②中断和陷入处理；③有限的低级进程管理和调度；④低级I/O。

中断和陷入处理：微内核OS将与硬件紧密相关的那一小部分放入微内核，此时微内核的主要功能是捕获所发生的中断和陷入事件，并进行中断响应处理，在识别中断或陷入的事件后，再发送给相关的服务器来处理，故中断和陷入处理也应放入微内核。因此选择C。

6．相对于传统操作系统结构，采用微内核结构设计和实现操作系统有诸多好处，下列(　)是微内核结构的优点。

Ⅰ．使系统更高效　　Ⅱ．添加系统服务时，不必修改内核

Ⅲ．微内核结构没有单一内核稳定　　Ⅳ．使系统更可靠

A．Ⅰ、Ⅲ、Ⅳ　　B．Ⅰ、Ⅱ、Ⅳ

C．仅Ⅱ、Ⅳ　　D．仅Ⅰ、Ⅳ

- 微内核将最基础的功能和与硬件有关的操作均放在内核中，其他各种服务器均与硬件无关，因此具有扩展性高，灵活性强；可靠性好和安全性高等优点；
- 由于微内核处于核心地位，也易于实现客户和服务器之间、服务器和服务器之间的通信采用消息传递机制。

6．相对于传统操作系统结构，采用微内核结构设计和实现操作系统有诸多好处，下列(　)是微内核结构的优点。

Ⅰ．使系统更高效　　Ⅱ．添加系统服务时，不必修改内核

Ⅲ．微内核结构没有单一内核稳定　　Ⅳ．使系统更可靠

A．Ⅰ、Ⅲ、Ⅳ　　B．Ⅰ、Ⅱ、Ⅳ

C．仅Ⅱ、Ⅳ　　D．仅Ⅰ、Ⅳ

C【解析】微内核结构需要频繁地在管态和目态之间进行切换，操作系统的执行开销相对偏大，那些移出内核的操作系统代码根据分层的原则被划分成若干服务程序，它们的执行相互独立，交互都借助于微内核进行通信，影响了系统的效率，因此Ⅰ不是优点。

由于内核的服务变少，且一般来说内核的服务越少，内核越稳定，所以Ⅲ错误。

Ⅱ、Ⅳ正是微内核结构的优点。因此选择C。

7．下列关于操作系统结构的说法，正确的是(　)。

Ⅰ．当前广泛使用的Windows XP操作系统，采用的是分层式OS结构

Ⅱ．模块化的OS结构设计的基本原则是每一层都仅使用其底层所提供的功能和服务，这样就使系统的调试和验证都变得容易

Ⅲ．由于微内核结构能有效支持多处理机运行，故非常适合于分布式系统环境

Ⅳ．采用微内核结构设计和实现操作系统具有诸多好处，如添加系统服务时，不必修改内核，使系统更高效

A．Ⅰ和Ⅱ　　B．Ⅰ和Ⅲ　　C．仅Ⅲ　　D．Ⅲ和Ⅳ

C【**解析**】Windows是微内核操作系统，Ⅰ错误。Ⅱ描述的是层次化构架的原则。

7．下列关于操作系统结构的说法，正确的是(　)。

Ⅰ．当前广泛使用的Windows XP操作系统，采用的是分层式OS结构

Ⅱ．模块化的OS结构设计的基本原则是每一层都仅使用其底层所提供的功能和服务，这样就使系统的调试和验证都变得容易

Ⅲ．由于微内核结构能有效支持多处理机运行，故非常适合于分布式系统环境

Ⅳ．采用微内核结构设计和实现操作系统具有诸多好处，如添加系统服务时，不必修改内核，使系统更高效

A．Ⅰ和Ⅱ　　B．Ⅰ和Ⅲ　　C．仅Ⅲ　　D．Ⅲ和Ⅳ

C【**解析**】在微内核构架中，客户和服务器之间、服务器和服务器之间的通信采用消息传递机制，这就使得微内核系统能很好地支持分布式系统，Ⅲ正确。

7．下列关于操作系统结构的说法，正确的是(　)。

Ⅰ．当前广泛使用的Windows XP操作系统，采用的是分层式OS结构

Ⅱ．模块化的OS结构设计的基本原则是每一层都仅使用其底层所提供的功能和服务，这样就使系统的调试和验证都变得容易

Ⅲ．由于微内核结构能有效支持多处理机运行，故非常适合于分布式系统环境

Ⅳ．采用微内核结构设计和实现操作系统具有诸多好处，如添加系统服务时，不必修改内核，使系统更高效

A．Ⅰ和Ⅱ　　B．Ⅰ和Ⅲ　　C．仅Ⅲ　　D．Ⅲ和Ⅳ

C【**解析**】添加系统服务时不必修改内核，这就使得微内核构架的可扩展性和灵活性更强。微内核构架的主要问题是性能问题，“使系统更高效”显然错误，Ⅳ错误。因此选择C。

8．采用微内核结构时，将操作系统分为用于实现基本功能的内核和提供各种服务的服务器两部分，通常必须包含在操作系统内核中的是(　)。

A．内存分配　　B．中断处理

C．文件处理　　D．命令处理

内核的功能包括：

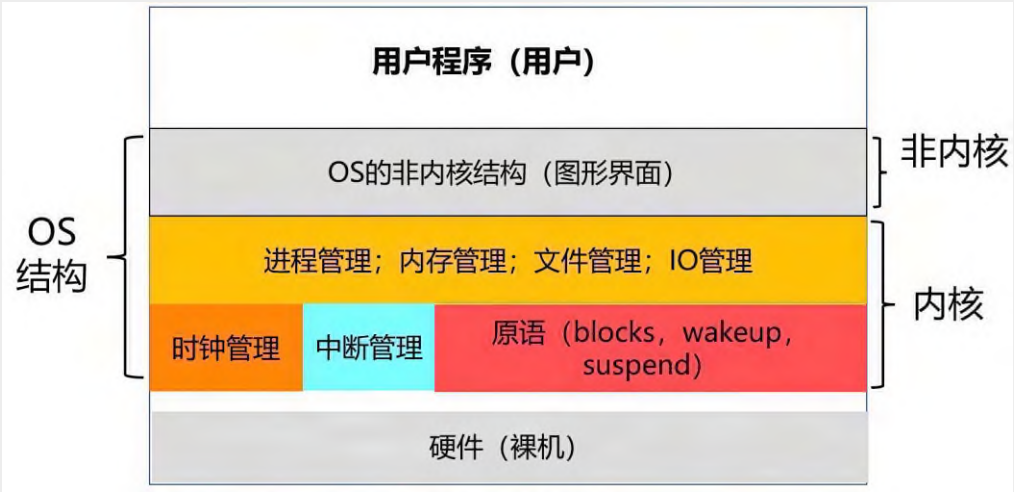
（1）I/O指令、置终端屏蔽指令、清内存、建存储保护、设置时钟指令。

（2）中断、异常、陷入，比如缺页中断等

（3）进程（线程）管理

（4）系统调用，比如调用了设备驱动程序

（5）用户内存地址的转换（逻辑---> 物理映射）



8．采用微内核结构时，将操作系统分为用于实现基本功能的内核和提供各种服务的服务器两部分，通常必须包含在操作系统内核中的是(　)。

A．内存分配　　B．中断处理

C．文件处理　　D．命令处理

B【解析】本题考查微内核的基本功能，包括进程（线程）的管理、低级存储器管理及中断和陷入处理。因此选择B。

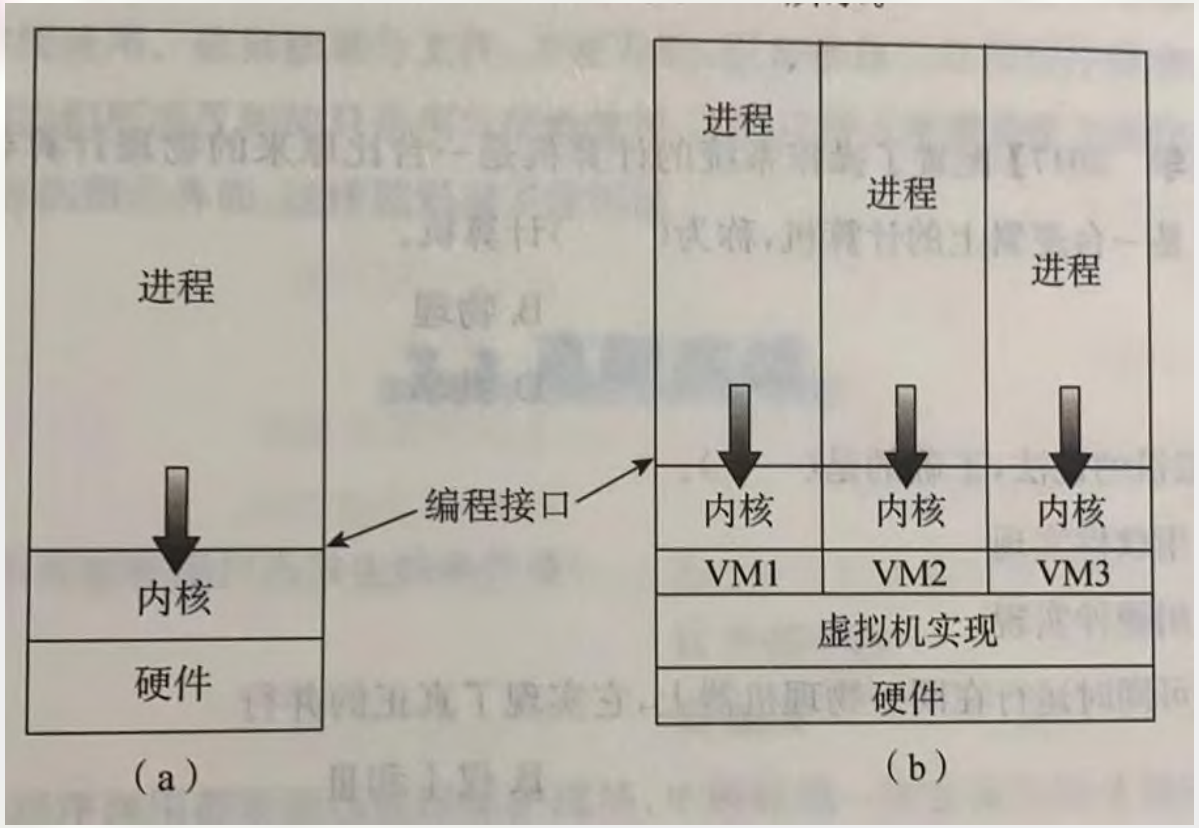
# 谢谢大家

# 考点五：虚拟机

# 考点五：虚拟机

- 虚拟机的基本思想是单个计算机(CPU、内存、磁盘、网卡等）的硬件抽象为几个不同的执行部件，从而造成一种“幻觉”,仿佛每个独立的执行环境都在自己的计算机上运行一样
- 通过利用CPU调度和虚拟内存技术，操作系统能带来一种“幻觉”，即进程认为有自己的处理器和自己的(虚拟)内存。当然，进程通常还有其他特征，如系统调用和文件系统，这些是硬件计算机所不能提供的。
- 虚拟机方法除了提供了与基本硬件相同的接口之外，并不提供额外功能。

# 考点五：虚拟机




- 每个进程都有一个与基本计算机一样的（虚拟）副本。

# 考点五：虚拟机

创建虚拟机有几个原因，最根本的是，

- 在并行运行几个不同的执行环境（即不同的操作系统）时能够共享相同的硬件。
- 虚拟机方法的主要困难与磁盘系统有关。假设物理机器有3个磁盘驱动器但是要提供7个虚拟机。它不能为每个虚拟机分配一个磁盘驱动器，因为虚拟机软件本身需要一定的磁盘空间以提供虚拟内存。

# 考点五：虚拟机

- 虚拟机的理念具有许多优点：
- 每个虚拟机完全独立于其他虚拟机，因此没有安全问题，但同时也没有直接资源共享。
- 提供共享有两种实现方法。第一，可以通过共享小型磁盘来共享文件，:这种方案模拟了共享物理磁盘，但通过软件实现。
- 第二，可以通过定义一个虚拟机的网络，每台虚拟机通过虚拟通信网络来传递消息。同样，该网络是按物理通信网络来模拟的，但是通过软件实现的。这样的虚拟机系统是用于研究和开发操作系统的好工具。

# 考点五：虚拟机

- 虚拟机的缺点：
  - 管理成本高
  - 系统性能下降（1+1 < 2）

【牛刀小试】

1．【广东工业大学　2017】配置了操作系统的计算机是一台比原来的物理计算机功能更强大的计算机，这样的计算机只是一台逻辑上的计算机，称为()计算机。

A．虚拟　　B．物理

C．并行　　D．共享

A【解析】配置了操作系统的计算机是一台比原来的物理计算机功能更强大的计算机，这样的计算机只是一台逻辑上的计算机，称为虚拟计算机。因此选择A。

2．下列关于虚拟机的说法，正确的是(　)。

Ⅰ．虚拟机可以用软件实现

Ⅱ．虚拟机可以用硬件实现

Ⅲ．多台虚拟机可同时运行在同一物理机器上，它实现了真正的并行

A．仅Ⅰ和Ⅱ　　B．仅Ⅰ和Ⅲ

C．仅Ⅰ　　D．Ⅰ、Ⅱ和Ⅲ

A【**解析**】虚拟机可以用软件实现；软件能实现的功能也能由硬件实现，所以虚拟机可以用硬件实现；

可以在同一物理机器上同时运行多台虚拟机，但虚拟机并不能实现真正的并行，实现真正并行的是多核处理机。因此选择A。

3．下列关于ＶMware Workstation虚拟机的说法，错误的是(　　)。

A．真实硬件不会直接执行虚拟机中的敏感指令

B．虚拟机中只能安装一种操作系统

C．虚拟机是运行在计算机中的一个应用程序

D．虚拟机文件封装在一个文件夹中，并存储在数据存储器中

B【**解析**】真实硬件不会直接执行虚拟机中的敏感指令；

虚拟机中可以安装多种操作系统；

虚拟机是运行在计算机中的一个应用程序；

虚拟机文件封装在一个文件夹中，并存储在数据存储器中。因此选择B。

4．(　)不是虚拟性在计算机中的应用。

A．多用户的分时操作系统　B．SPOOLing系统

C．虚拟存储器　　D．交换与覆盖

D【**解析**】操作系统用到了很多虚拟技术，其中一个就是虚拟存储。为了提高内存利用率，进程并不是将所有代码都装入内存，而是部分装入，即使是在内存足够大的情况下，有时当进程离开运行态时会被对换到虚拟内存中，也就是硬盘的一部分。

SPOOLing技术也是操作系统虚拟性的一个实现，为了减少等待和请求的重复申请，提高效率，允许硬件设备虚拟为很多台虚拟设备实现脱机工作。有些分时操作系统更是将计算机虚拟成多台计算机供多个用户同时使用，当然，是并发的而不是并行的，这也是虚拟性。

覆盖与交换技术是在多道程序环境下用来扩充内存的两种方法。因此选择D。

# 谢谢大家

# 【真题实战】

1、下列选项中，不可能在用户态发生的事件是_____。

A. 系统调用　　B. 外部中断

C. 进程切换　　D. 缺页

| | 发生位置 | 处理位置 |
|---|---|---|
| 中断 | 用户态/内核态 | 内核态 |
| 异常 | 用户态/内核态 | 内核态 |
| 缺页 | 用户态/内核态 | 内核态 |
| 系统调用 | 用户态 | 内核态 |
| 进程创建 | 用户态/内核态 | 内核态 |
| 进程切换 | 内核态 | 内核态 |
| 进程调度 | 内核态 | 内核态 |

C 【解析】根据上表可以得知，进程切换不可能在用户态发生。

2、中断处理和子程序调用都需要压栈以保护现场，中断处理一定会保存而子程序调用不需要保存其内容的是_____。

A. 程序计数器　　B. 程序状态字寄存器

C. 通用数据寄存器　　D. 通用地址寄存器

B【解析】中断处理和子程序调用的区别主要有

（1）程序是否提前安排好：中断服务程序是随机的，而普通子程序是预先安排好的。

（2）结束程序不同：中断服务子程序以RETI结束，而一般子程序以RET结束。

（3）结束动作不同：中断服务子程序RETI除将断点弹回PC动作外，还要清除对应的中断优先标志位，以便新的中断请求能被响应。一般子程序则无此项操作。

2、中断处理和子程序调用都需要压栈以保护现场，中断处理一定会保存而子程序调用不需要保存其内容的是_____。

A. 程序计数器　　B. 程序状态字寄存器

C. 通用数据寄存器　　D. 通用地址寄存器

B【解析】中断处理和子程序调用都需要压栈以保护现场，这个现场包括PC（为了保证能够准确找到断点）、通用数据寄存器和通用地址寄存器（在执行中断服务程序和子程序时，这些可能会发生变化）。

但是在中断处理中，中断服务程序和当前程序是无关的，因此还需要保存当前程序的状态信息，即PSWR中的信息，而对于子程序调用而言，除了PSW因为和父程序内容一致而不需要保存，因此B选项正确。

3、内部异常（内中断）可分为故障（fault）、陷阱（trap）和终止（abort）三类。下列有关内部异常的叙述中，错误的是_____。

A．内部异常的产生与当前执行指令相关

B．内部异常的检测由 CPU 内部逻辑实现

C．内部异常的响应发生在指令执行过程中

D．内部异常处理后返回到发生异常的指令继续执行

D【解析】内中断是指来自CPU内部产生的中断，包括程序运算引起的各种错误，如地址非法、检验错、访存缺页、存储访问控制错、算术操作上溢、数据格式非法、除数为0、非法指令、用户程序执行特权指令、用户态到核心态的切换、运行超时、等待超时、算术运算错等，以上都是在指令的执行过程中产生的，故A正确。

3、内部异常（内中断）可分为故障（fault）、陷阱（trap）和终止（abort）三类。下列有关内部异常的叙述中，错误的是_____。

A．内部异常的产生与当前执行指令相关

B．内部异常的检测由 CPU 内部逻辑实现

C．内部异常的响应发生在指令执行过程中

D．内部异常处理后返回到发生异常的指令继续执行

D【解析】这种检测异常的工作是由CPU(包括控制器和运算器)实现的，故B正确。内中断属于非屏蔽中断，一旦出现应立即处理，否则程序也无法继续执行，C正确。

3、内部异常（内中断）可分为故障（fault）、陷阱（trap）和终止（abort）三类。下列有关内部异常的叙述中，错误的是_____。

A．内部异常的产生与当前执行指令相关

B．内部异常的检测由 CPU 内部逻辑实现

C．内部异常的响应发生在指令执行过程中

D．内部异常处理后返回到发生异常的指令继续执行

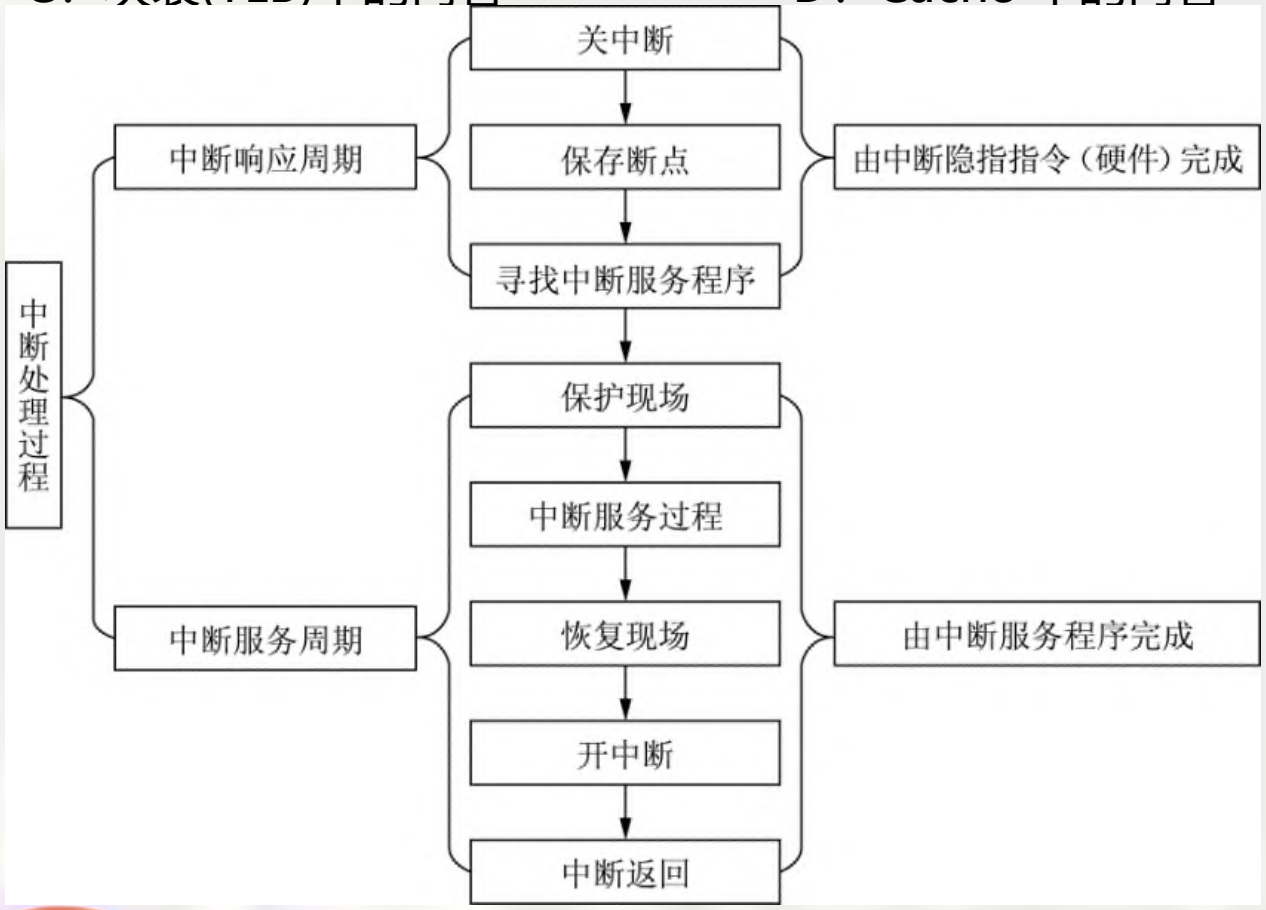
D【解析】对于一般的异常，中断处理后，不会再返回到发生异常的指令处继续执行。例如除数为0引起异常，再次回到该指令，还是会再次引发异常，因此D错误。

- 页缺失是特殊的内中断，其特殊性包括两个，第一个是指令执行过程中可能会发生多次中断，如一条读取数据的多字节指令，指令本身跨越两个页面，若指令后一部分所在页面和数据所在页面均不在内存，则该指令的执行至少产生两次缺页中断。第二个是中断处理完成后，将会回到被中断指令处继续执行。

4、处理外部中断时，应该由操作系统保存的是_____。

A．程序计数器(PC)的内容　　B．通用寄存器的内容

C．块表(TLB)中的内容　　D．Cache 中的内容



4、处理外部中断时，应该由操作系统保存的是_____。

A．程序计数器(PC)的内容　　B．通用寄存器的内容

C．块表(TLB)中的内容　　D．Cache 中的内容

B【解析】中断过程包括中断响应周期和中断处理周期，中断响应周期由中断隐指令（硬件）完成，实现关中断、保存断点（PC和PSWR）、中断服务程序送PC，因此A不是操作系统保存的。

中断处理周期由中断服务程序（软件）完成，包括保护现场（通用寄存器）、中断服务、恢复现场、开中断和中断返回，因此B正确。

快表和Cache都是由硬件来保存的，对操作系统是透明的，C、D错误。

5、假定下列指令已装入指令寄存器。则执行时不可能导致 CPU 从用户态变为内核态(系统态)的是_____。

A．DIV R0,R1 ; (R0)/(R1)→R0

B．INT n ; 产生软中断

C．NOT R0 ; 寄存器 R0 的内容取非

D．MOV R0,addr ; 把地址 addr 处的内存数据放入寄存器 R0 中

用户态向内核态的转换

- **系统调用**：这是用户态进程主动要求切换到内核态的一种方式，用户态进程通过系统调用申请使用操作系统提供的服务程序完成工作，而系统调用的机制其核心还是使用了操作系统为用户特别开放的一个中断来实现。
- **异常**：当CPU在执行运行在用户态下的程序时，发生了某些事先不可知的异常，这时会触发由当前运行进程切换到处理此异常的内核相关程序中，也就转到了内核态，比如缺页异常。
- **外围设备的中断**：当外围设备完成用户请求的操作后，会向CPU发出相应的中断信号，这时CPU会暂停执行下一条即将要执行的指令转而去执行与中断信号对应的处理程序，如果先前执行的指令是用户态下的程序，那么这个转换的过程自然也就发生了由用户态到内核态的切换。

5、假定下列指令已装入指令寄存器。则执行时不可能导致 CPU 从用户态变为内核态(系统态)的是_____。

A．DIV R0,R1 ; (R0)/(R1)→R0

B．INT n ; 产生软中断

C．NOT R0 ; 寄存器 R0 的内容取非

D．MOV R0,addr ; 把地址 addr 处的内存数据放入寄存器 R0 中

C【解析】从用户态变为内核态(系统态)的是的途径是中断、异常和系统调用。

对于A选项，当R1是0时，会产生异常，因此A可能导致 CPU 从用户态变为内核态(系统态)。

对于B选项，产生软中断，可能导致 CPU 从用户态变为内核态(系统态)。

对于D选项，把地址 addr 处的内存数据放入寄存器 R0 中。

当addr处的数据不在内存中时，会产生缺页中断，因此可能导致 CPU 从用户态变为内核态(系统态)。选项C是寄存器 R0 的内容取非，仅仅是从0变为1或者1变为0，不可能产生中断，从因此C选项正确。

6、执行系统调用的过程包括如下主要操作：

①返回用户态　　②执行陷入(trap)指令

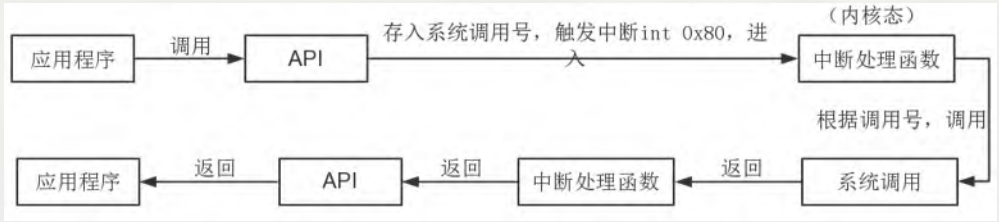
③传递系统调用参数　　④执行相应的服务程序

正确的执行顺序是_____。

A．②→③→①→④　　B．②→④→③→①

C．③→②→④→①　　D．③→④→②→①

系统调用的过程是一个比较复杂的过程，如所示，请注意应用程序通过调动API，发起系统调用，此时程序处于用户，其执行过程是，

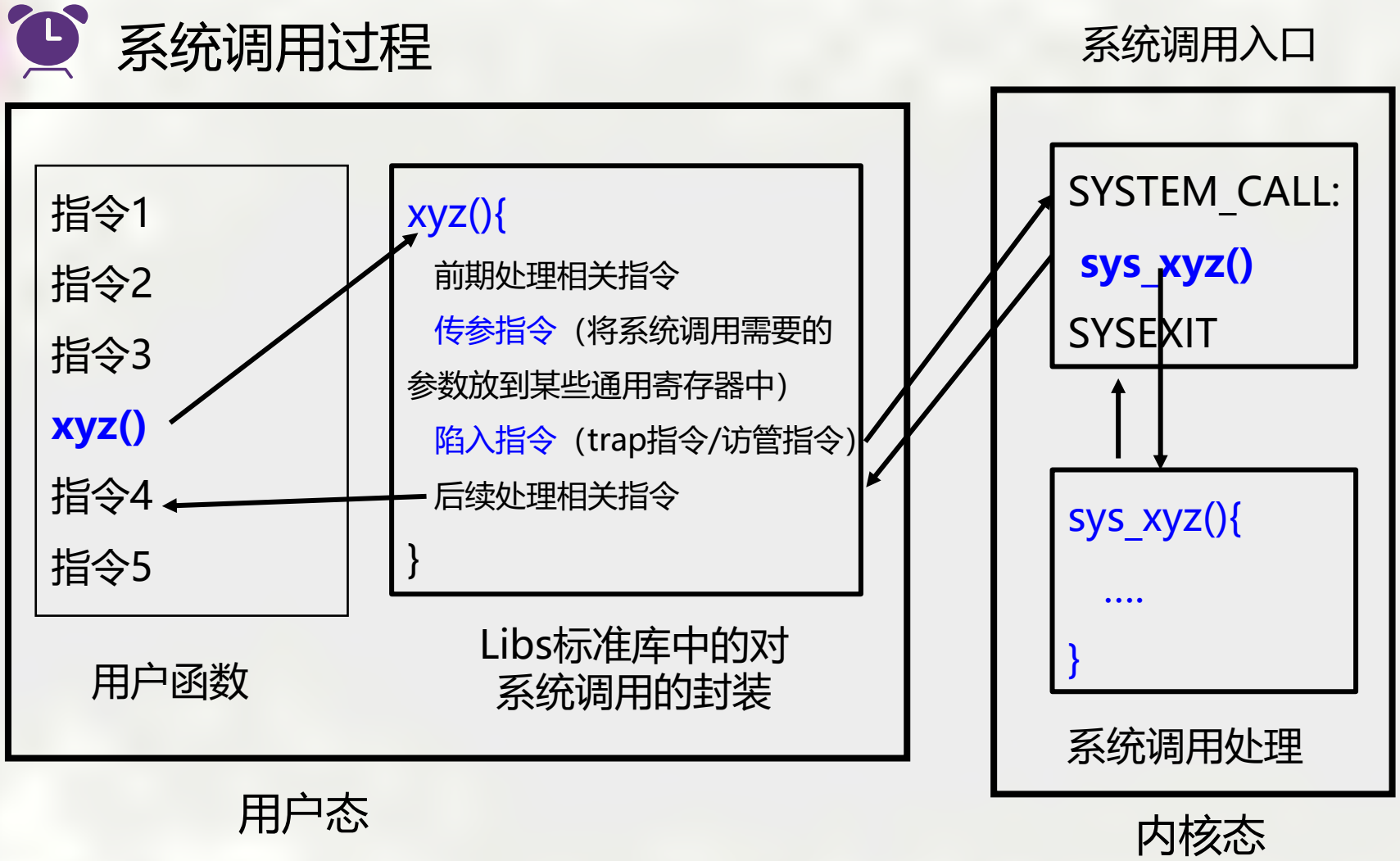



①当系统调用发生时，处理器通过一种特殊的机制（此处的中断编号是int 0x80），通常是中断或者异常处理，把控制流程转移到内核态的一些特定的位置，在该过程中，会传递系统调用参数，然后由陷入 (trap) 指令负责将用户态转化为内核态。

②由内核程序执行被请求的功能代码，此时CPU 执行相应的内核态服务程序。这个功能代码代表着对一段标准程序段的执行，用以完成所请求的功能。因此在执行系统调用服务程序的过程中，CPU 处于内核态。

③处理结束之后，程序状态恢复系统调用之前的现场；把运行模式从特权模式恢复成为用户方式。最后通过API将控制权转移回原来的用户程序。

# 系统调用过程


指令1
指令2
指令3
xyz()
指令4
指令5
用户函数
xyz(){
前期处理相关指令
传参指令（将系统调用需要的参数放到某些通用寄存器中）
陷入指令（trap指令/访管指令）
后续处理相关指令
}
Libs标准库中的对系统调用的封装
用户态
系统调用入口
SYSTEM_CALL:
sys_xyz()
SYSEXIT
sys_xyz(){
....
}
系统调用处理
内核态

6、执行系统调用的过程包括如下主要操作：

①返回用户态　　②执行陷入(trap)指令

③传递系统调用参数　　④执行相应的服务程序

正确的执行顺序是_____。

A．②→③→①→④　　B．②→④→③→①

C．③→②→④→①　　D．③→④→②→①

C【解析】因此其过程是传递系统调用参数、执行陷入(trap)指令、执行相应的服务程序和返回用户态，选择C、

7、计算机系统中有些操作必须使用特权指令完成,下面()操作不需要使用特权指令。

A. 修改界限地址寄存器　　B.设置定时器初值

C. 触发访管指令　　D.关闭中断允许位

C【解析】特权指令是指在内核态执行的指令,只允许操作系统使用,不允许应用程序使用。A、B、D三项操作都需要使用特权指令。

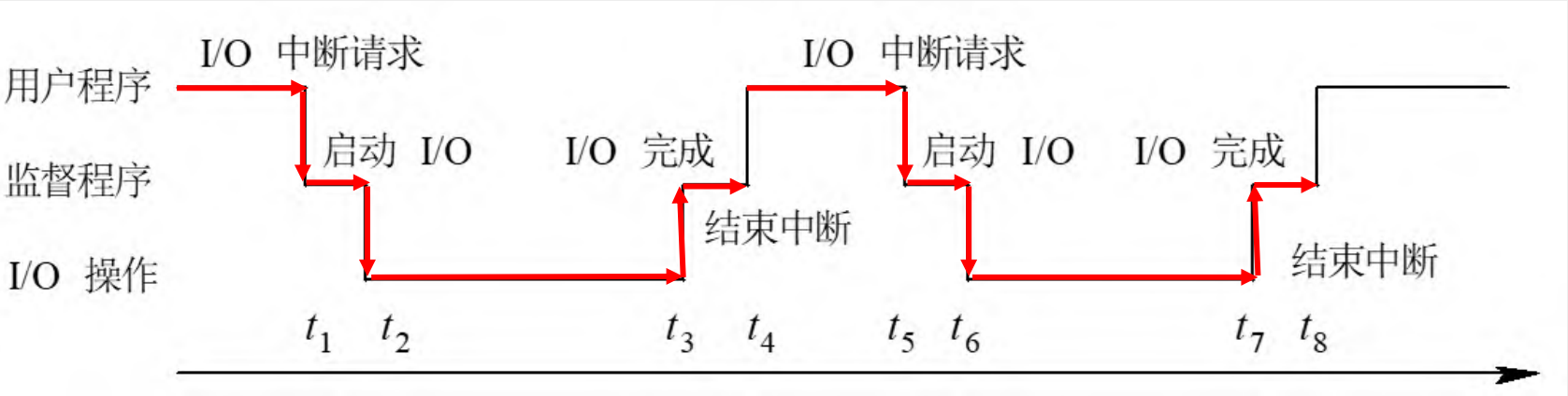
非特权指令是指在用户态执行的指令,允许应用程序使用。访管指令是可以在用户态执行的指令,触发访管指令操作不需要使用特权指令。

8、单道批处理系统的主要缺点是()。

A. CPU的利用率不高　　B.失去了交互性

C. 不具备并行性　　D.以上都不是

A【解析】单道批处理系统的内存中只有一道程序,当该程序发出IO请求后,CPU必须等待IO完成,又因IO设备的低速性,就会使CPU长时间处于空闲状态,导致CPU的利用率不高。



9、系统调用的目的是()。

A．请求系统服务　　B.终止系统服务

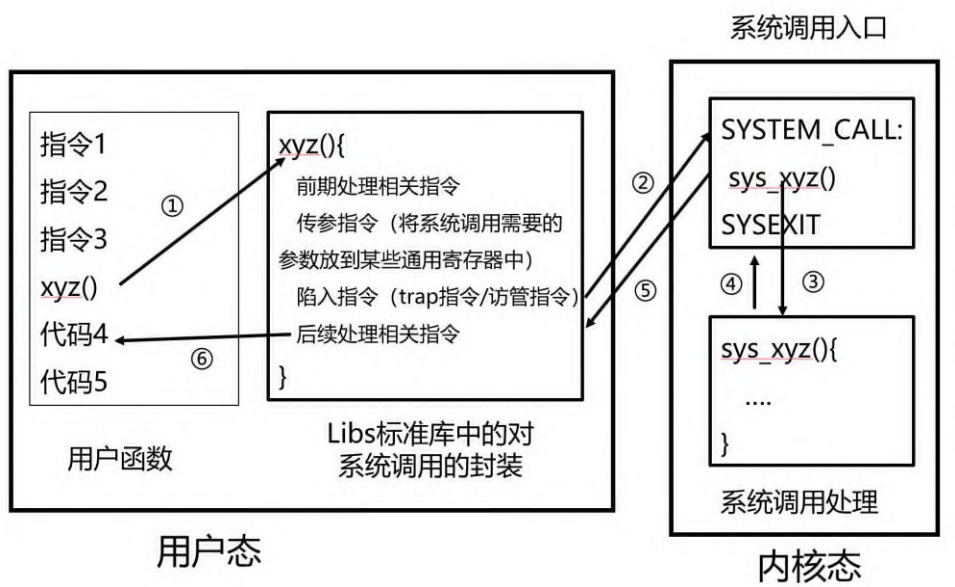
C.　申请系统资源　　D.释放系统资源

A【解析】系统调用的目的是使应用程序能够通过系统调用间接调用操作系统中的相关过程,获取相应的系统服务。

10、程序设计时需要调用操作系统提供的系统调用,被调用的系统调用命令经过编译后，形成若干参数,并( )。

A. 访管指令或软中断　　B．启动I/O指令

C. 屏蔽中断指令　　D. 通道指令




A【解析】用户程序设计时，使用系统调用命令，该命令经过编译后，形成若干参数和访管指令或软中断

11、多道程序设计能充分发挥()之间的并行工作能力

A. CPU与外设　　B．进程与进程

C. 内存与进程　　D. 内存与外设

A【解析】采用多道程序设计技术后,CPU与外设能并行工作,从而提高资源利用率和系统吞吐量。

# 谢谢大家