



# 操作系统

Operating system

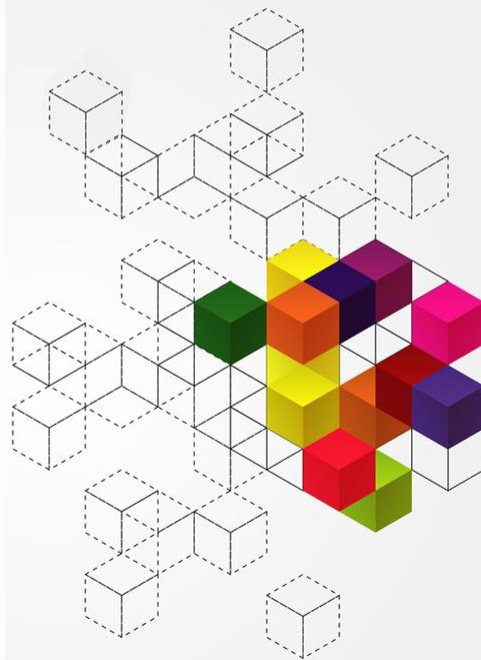
胡燕

大连理工大学

# 操作系统思维导图

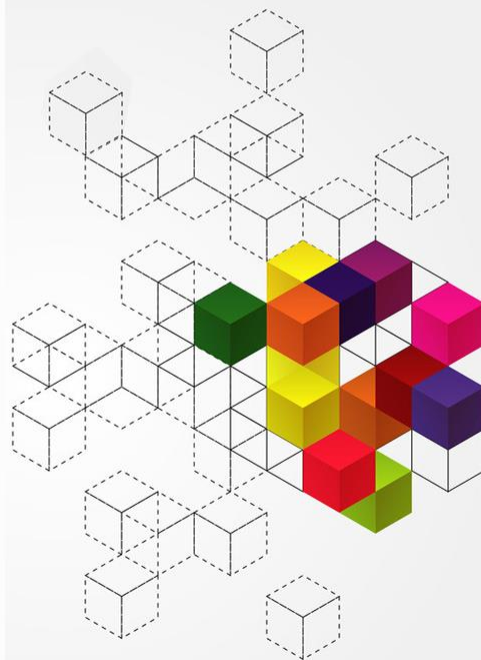
## 参考网址

- <https://blog.csdn.net/huzai9527/article/details/85283630>



# 操作系统Part 1 绪论

- 操作系统的定义
- 操作系统的功能
- 操作系统的基本类型及特点

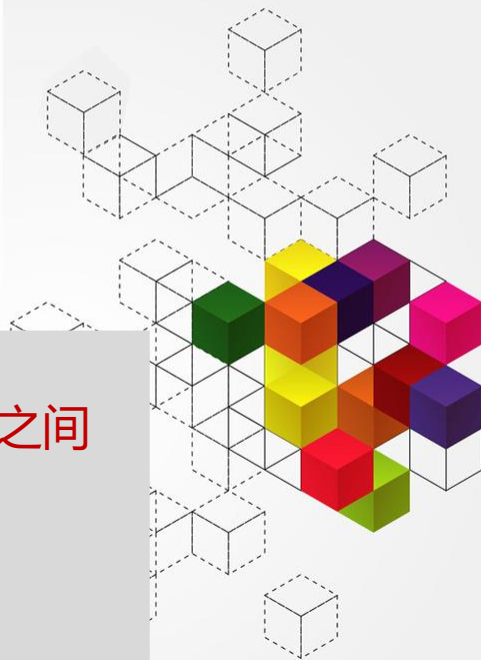


# 操作系统Part 1 绪论

- 操作系统的定义
- 操作系统的功能
- 操作系统的基本类型及特点

## • 1.操作系统的定义

- 操作系统是一种系统软件，介乎计算机用户与计算机硬件之间
- 操作系统管理硬件，并向上服务用户



# 操作系统Part 1 绪论

- 操作系统的定义
- 操作系统的功能
- 操作系统的基本类型及特点

- 2.操作系统的功能

- OS的职能是管理和控制计算机系统的所有软硬件资源，合理的组织计算机工作流程，并为用户提供一个良好的工作环境与友好的接口。
- OS基本功能模块包括：处理机管理、存储管理、设备管理、信息管理（文件系统管理）、用户接口。



# 操作系统Part 1 绪论

- 操作系统的定义
- 操作系统的功能
- 操作系统的基本类型及特点

## • 3.操作系统基本类型和特点

常见的几类操作系统包括：

**批处理系统**：操作员把用户提交的作业分类，把一批作业编成一个作业执行序列，由监控程序控制自动依次执行。特点：用户脱机使用、交互性差、成批处理、多道程序运行。

**分时系统**：把处理机的运行时间分成很短的时间片，按时间片轮转的方式，把处理机分配给各进程使用。特征：交互性、多用户同时性、独立性。

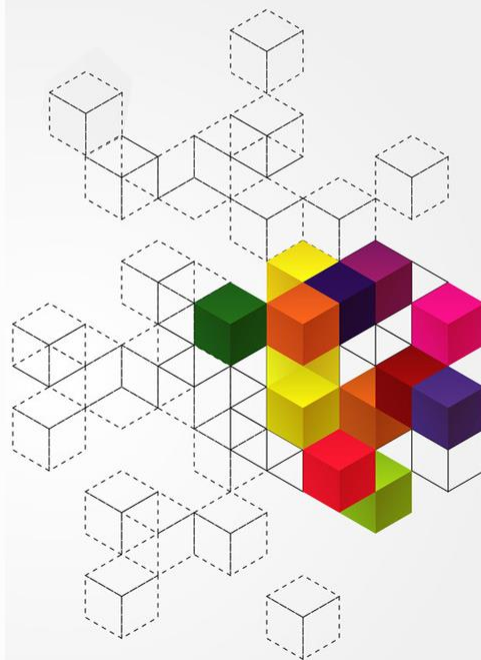
**实时系统**：对随机发生的外部事件作出及时响应并对其进行处理。特征：对关键事件的实时响应能力强、可靠性要求高、资源利用率低。

你还可以了解更多一些...（如今的领域细分，产生物联网操作系统、ROS等）



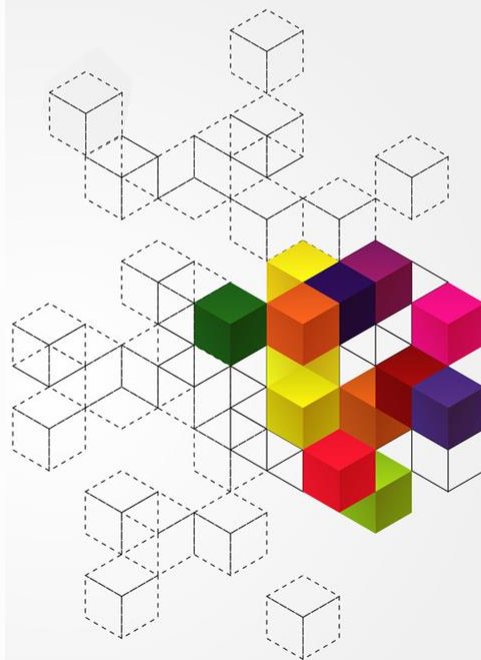
## 操作系统Part 2 操作系统结构

- 操作系统组成
- 系统模式和用户模式
- 系统调用及处理过程
- 操作系统结构



# 操作系统Part 2 操作系统结构

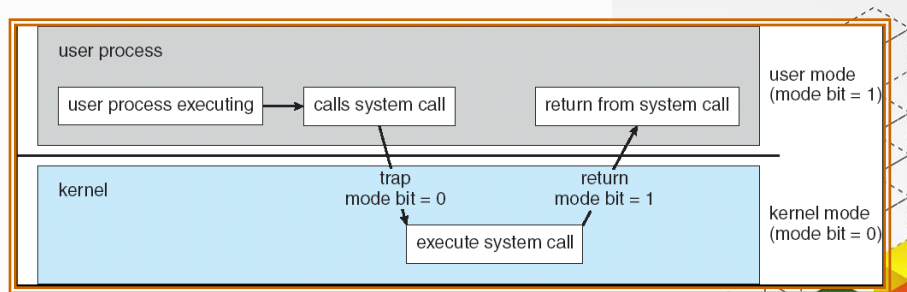
- 操作系统组成
  - 系统模式和用户模式
  - 系统调用及处理过程
  - 操作系统结构
- 操作系统主要由一下几个重要的模块构成
    - 进程管理
    - 存储管理
    - 文件管理
    - 设备管理
    - 用户界面



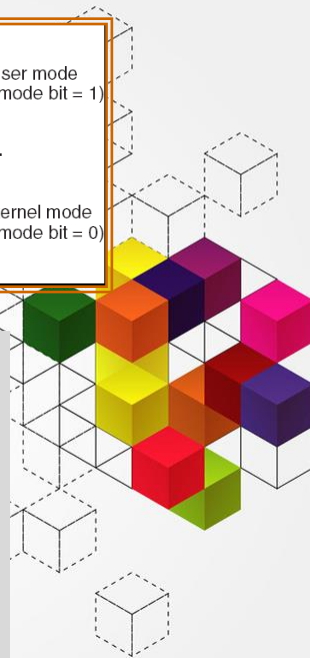


# 操作系统Part 2 操作系统结构

- 操作系统组成
- 系统模式和用户模式
- 系统调用及处理过程
- 操作系统结构



- OS通过两种执行模式（Dual-Mode）来对核心代码实施保护
  - 用户程序代码执行在用户态（User mode）
  - 操作系统核心代码执行在核心态（kernel mode）
  - 两种状态通过硬件的一个模式位（Mode bit）来表示



# 操作系统Part 2 操作系统结构

- 操作系统组成
- 系统模式和用户模式
- 系统调用及处理过程
- 操作系统结构

- 系统调用概念

- 系统调用是操作系统为上层应用提供的最基本的服务接口（界面）
- 当应用程序需要访问特权指令时，必须经过系统调用接口进入内核访问

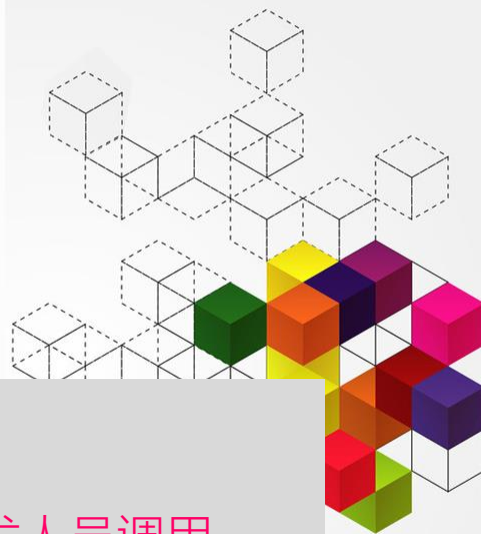


## 操作系统Part 2 操作系统结构

- 操作系统组成
- 系统模式和用户模式
- 系统调用及处理过程
- 操作系统结构

- 系统调用 vs. 应用编程接口 (API)

- 系统调用是操作系统为上层应用提供的最原始接口
- API是在用户态代码库中实现一些常规的基本功能，供开发人员调用
- API有可能使用到系统调用，也可能不使用系统调用

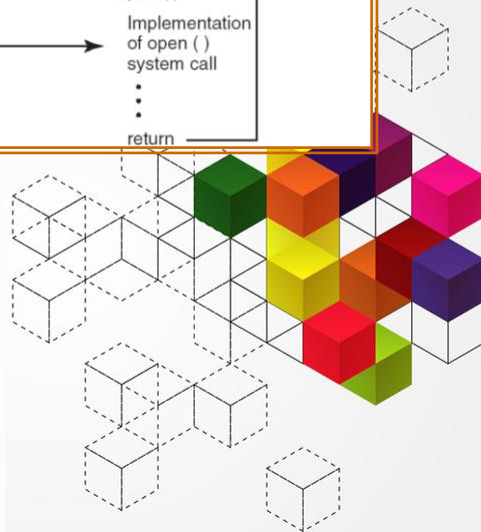
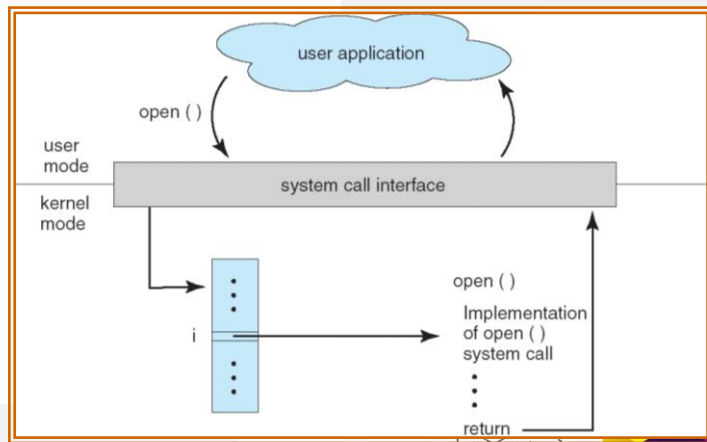


# 操作系统Part 2 操作系统结构

- 操作系统组成
- 系统模式和用户模式
- 系统调用及处理过程
- 操作系统结构

## • 系统调用的处理

- 用户在需要操作系统服务时，调用系统调用，陷入内核（在调用系统调用陷入内核时，会向OS内核传入一个系统调用号*i*）
- 进入内核后，根据*i*查找系统调用表，找到编号为*i*的系统调用的处理代码
- 内核执行完系统调用处理代码后，从核心态返回用户态



# 操作系统Part 2 操作系统结构

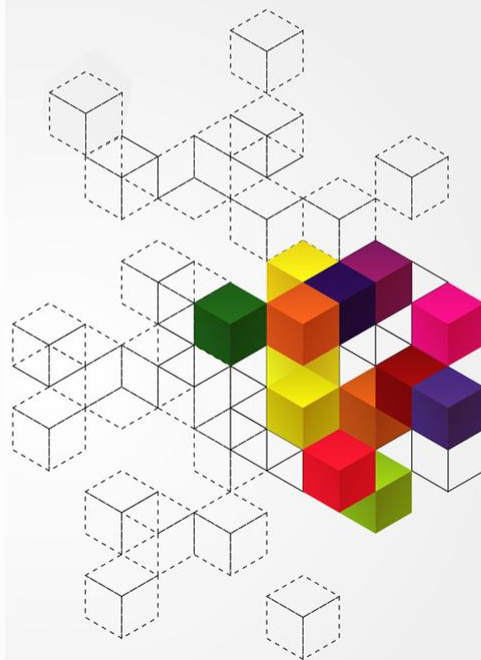
- 操作系统组成
- 系统模式和用户模式
- 系统调用及处理过程
- 操作系统结构

1.宏内核（单内核）

2.微内核

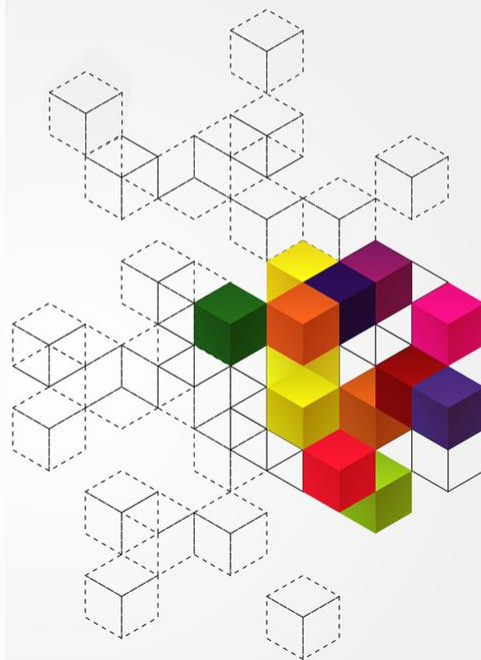
3.混合内核

要求：能用简单的语言说明不同结构OS的特点



# 操作系统Part 3 进程管理

- 并行和并发
- 进程的概念和特点
- 进程的组成和描述
- 进程上下文内容
- 进程上下文切换过程
- 进程空间和大小
- 进程的状态及转换



# 操作系统Part 3 进程管理

- 并行和并发

- 进程的概念和特点

- 进程的组成和描述

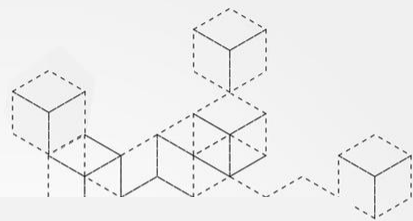
- 进程上下文内容

- 进程上下文切换过程

- 并行与并发

- 并发 (Concurrency) : 多个任务在同一时间段内执行。操作系统是一个并发系统, 各进程间的并发, 系统与应用进程间的并发。管理并发进程, 是操作系统的核心任务之一。

并行(parallelism): 多个任务在同一时刻进行处理。



# 操作系统Part 3 进程管理

- 并行和并发
- 进程的概念和特点
- 进程的组成和描述
- 进程上下文内容

## 进程与程序有何区别和联系？

- (1) 进程是一个**动态**概念，程序是一个**静态**概念。
- (2) 进程有**生命周期**，有诞生有消亡，短暂的；而程序在磁盘上持久存储
- (3) 进程是**竞争计算机系统资源**的基本单位，程序不是
- (4) 进程具有**并发特征**(独立性，异步性)，程序则没有。
- (5) 不同的进程可以是同一程序的不同执行。



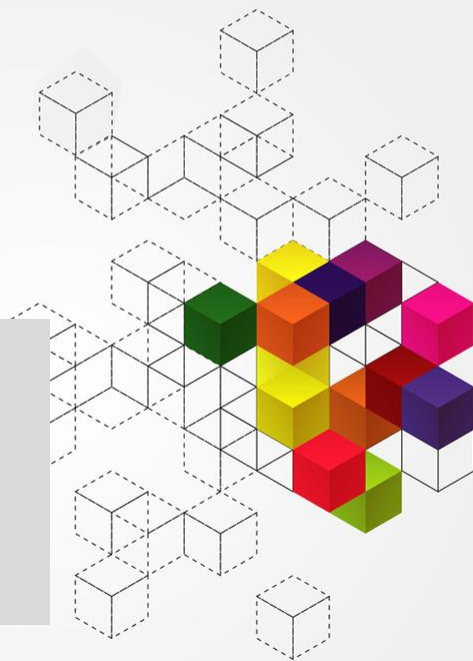


# 操作系统Part 3 进程管理

- 并行和并发
- 进程的概念和特点
- 进程的组成和描述
- 进程上下文内容
- 进程上下文切换过程
- 进程空间和大小

## 什么是线程？

- 线程是在进程内用于调度和占有处理机的基本单位
- 它由线程控制表、存储线程上下文的用户栈以及核心栈组成。



# 操作系统Part 3 进程管理

① 调度

② 并行性

③ 拥有资源

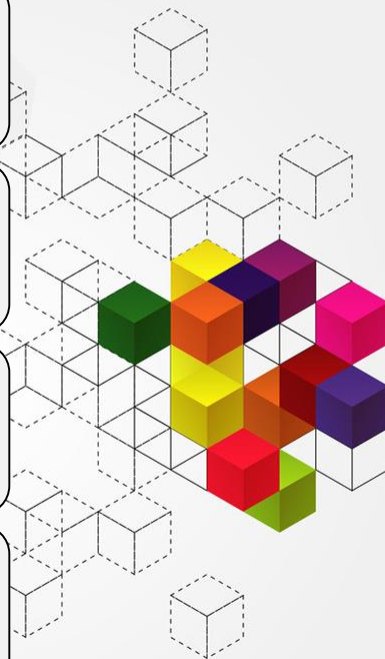
④ 系统开销

传统操作系统中，拥有资源的基本单位和独立调度分派的基本单位都是进程；而引入线程的操作系统中，线程是调度和分派的基本单位，进程则是资源分配的基本单位。

在引入线程的OS中，进程之间可以并发执行，同一进程的多个线程之间也可以并发执行，从而使得OS具有更好的并行性。

在OS中，进程是拥有资源的一个独立单位，它拥有自己的资源，而线程一般不拥有系统资源，但是它可以访问其隶属进程的资源。

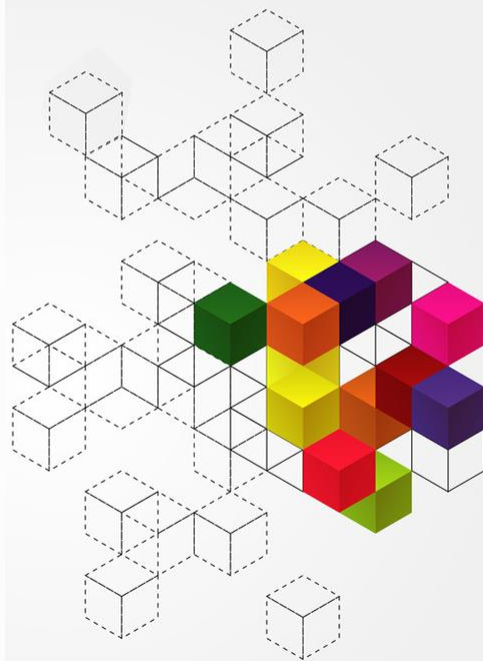
创建和撤销进程涉及资源的分配或回收，需要比线程创建和撤销大得多的系统开销，同样的，进程切换的开销也远远大于线程切换的开销。



# 操作系统Part 3 进程管理

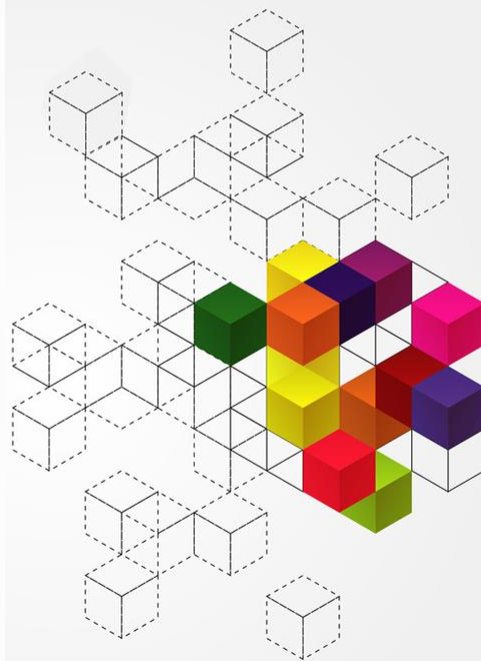
## 进程同步

- 临界区和临界资源的概念
- 进程的互斥含义
- 信号量的含义,P,V原语的含义
- 如何使用信号量完成互斥
- 如何使用信号量完成同步
- 理解经典同步问题



## 操作系统Part 3 进程管理

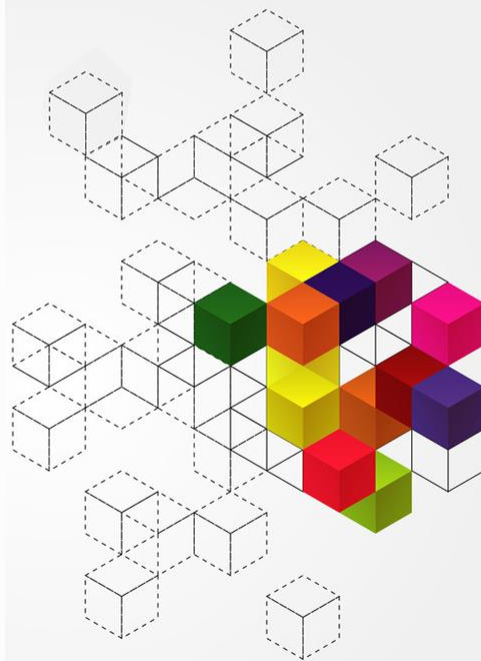
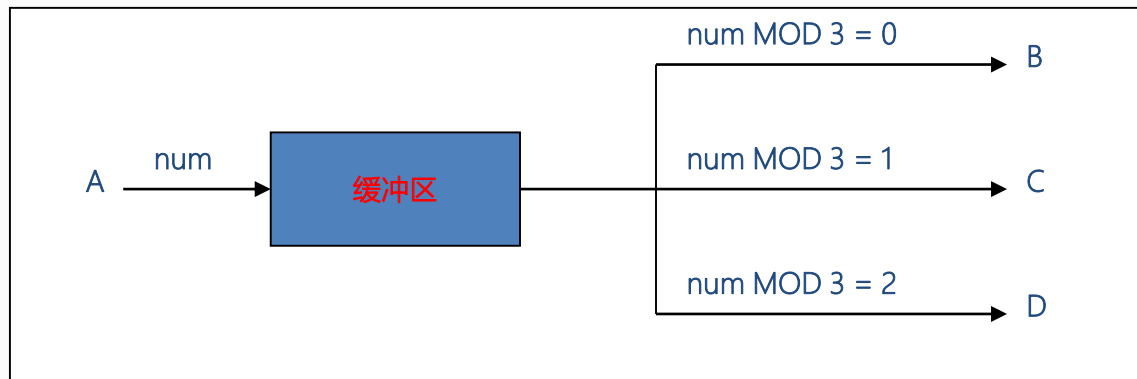
- **临界区**是指不允许多个并发进程交叉执行，一次最多允许一个进程进入的一段程序代码。
- 临界区是由于不同并发进程的程序段共享公用数据或公用数据变量而引起的。这些需要互斥访问的资源称为**临界区资源**。

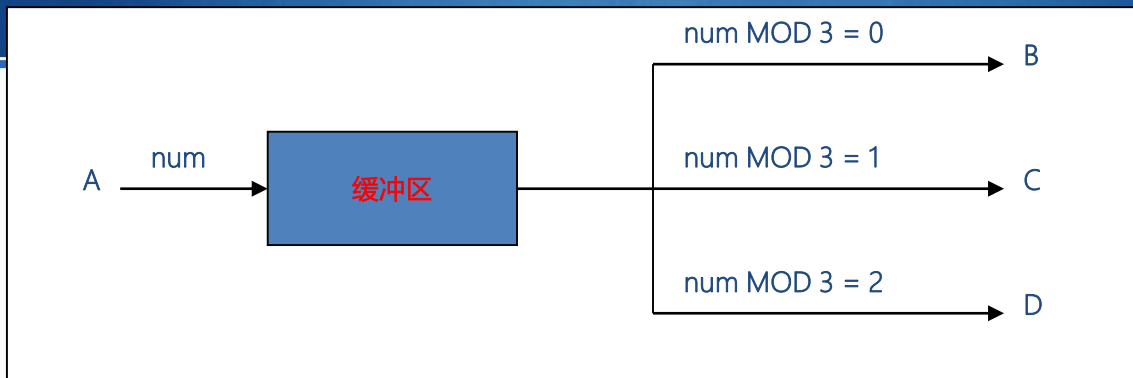


# 操作系统Part 3 进程管理

设有4个进程A、B、C、D共享一个缓冲区(大小为1),  
进程A负责循环地从文件读一个整数并放入缓冲区  
进程B从缓冲区循环读入MOD 3为0的整数并累计求和  
进程C从缓冲区循环地读入MOD 3为1的整数并累计求和  
进程D从缓冲区中循环地读入MOD 3为2的整数并累计求和。  
请用P、V操作写出能够正确执行的程序。

问题描述:





**Process PA**

```
Begin
P(Sempty);
<读入num至缓冲区>
if (num MOD 3 = 0)
    V(SB);
elif (num MOD 3 = 1)
    V(SC);
else
    V(SD);
end
```

**Process PB**

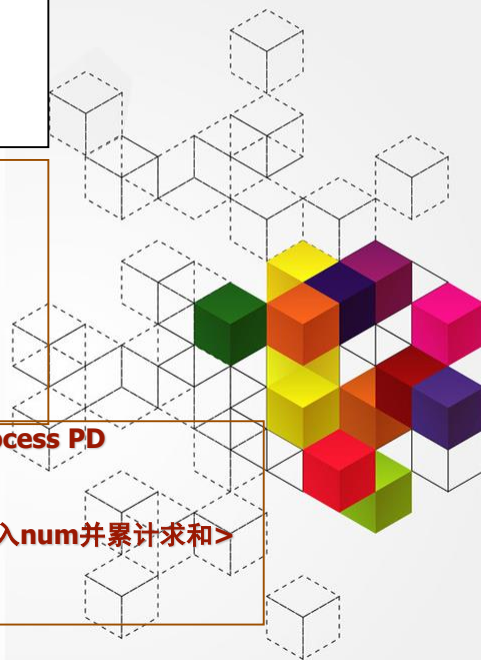
```
Begin
P(SB);
<从缓冲区读入num并累计求和>
V(Sempty);
end
```

**Process PC**

```
Begin
P(SC);
<从缓冲区读入num并累计求和>
V(Sempty);
end
```

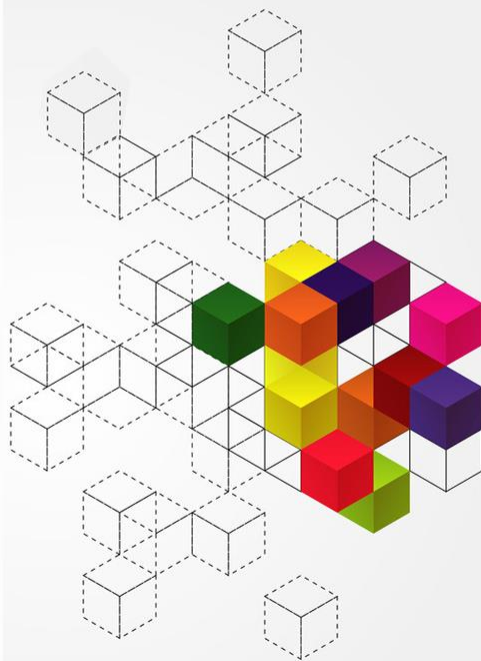
**Process PD**

```
Begin
P(SD);
<从缓冲区读入num并累计求和>
V(Sempty);
end
```

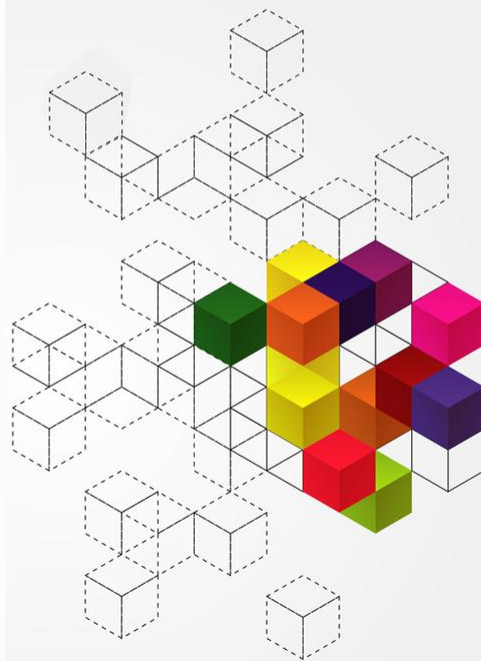
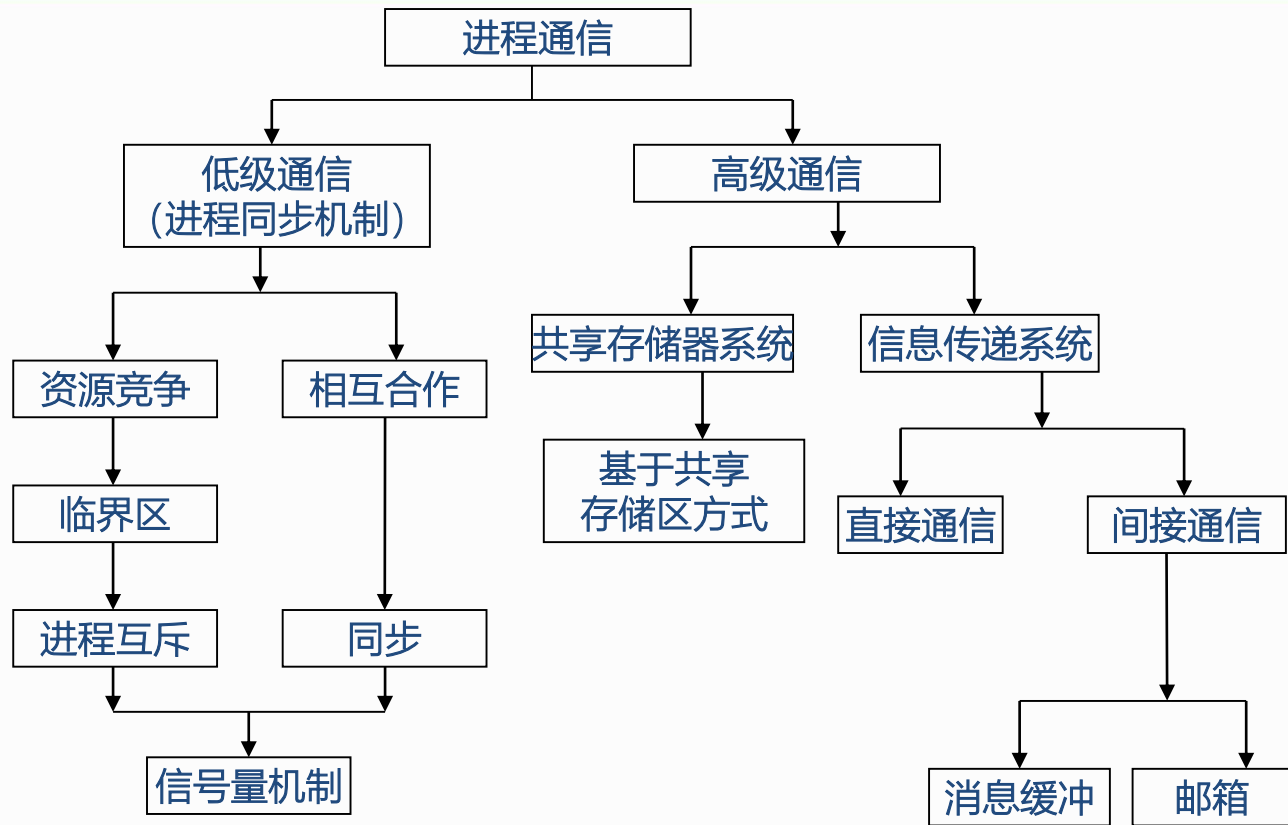


# 操作系统Part 3 进程管理

- 进程间通信方式有哪些?
- 死锁的概念
- 死锁产生根本原因和必要条件
- 死锁的排除方法
- 线程的概念及引入的原因
- 线程的分类

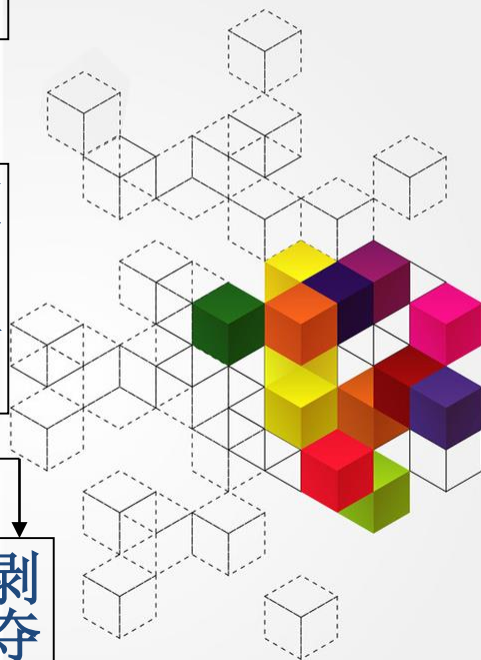
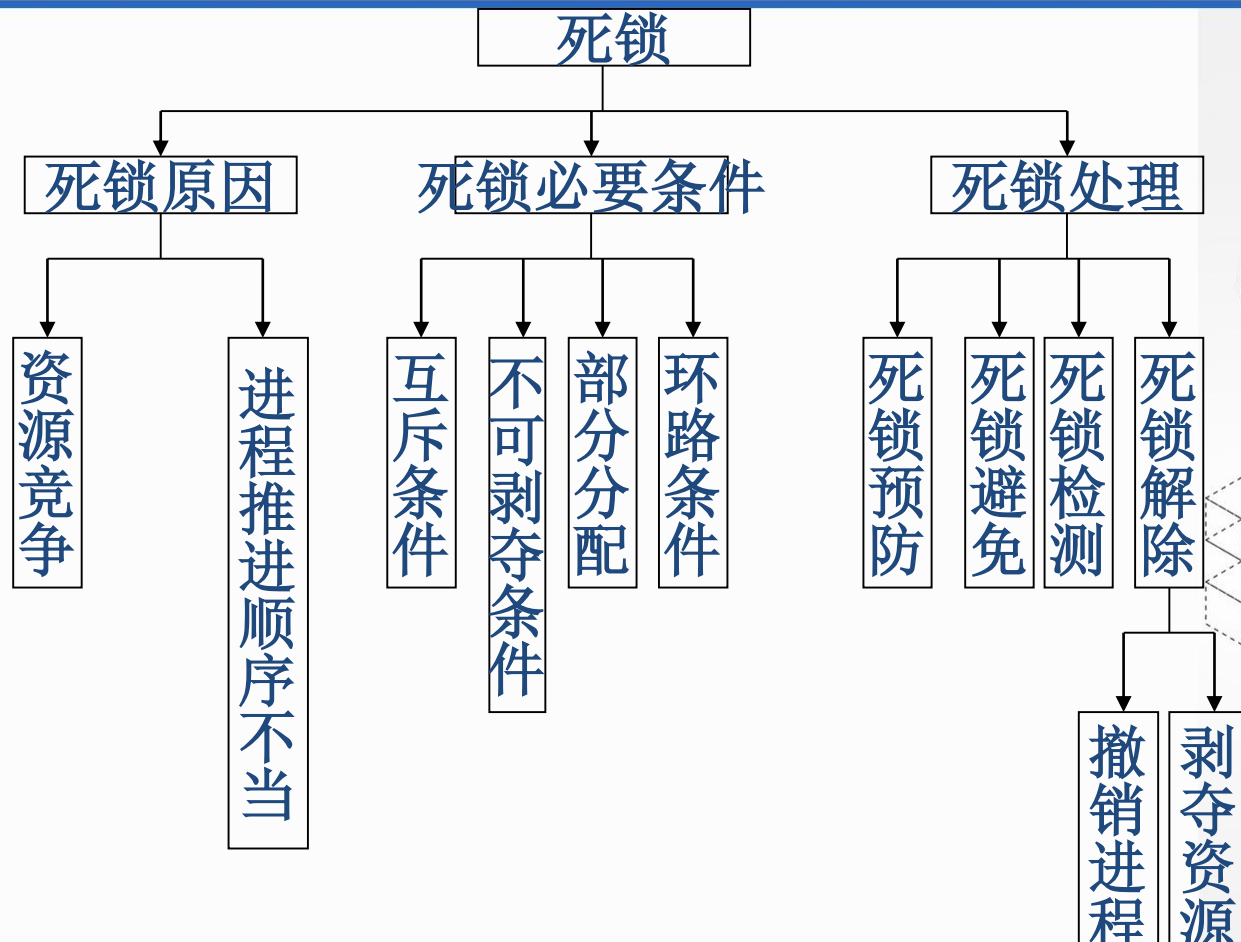


# 进程的同步与通信





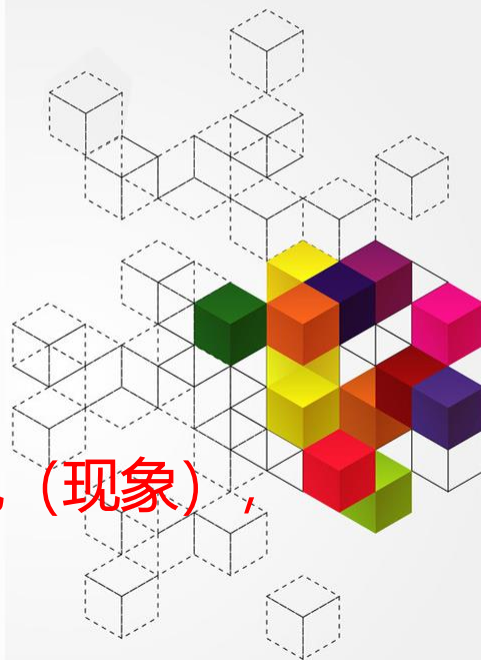
# 死锁



# 死锁习题1

- 产生死锁的基本原因是（ ）和（ ）。
- 答案：系统资源不足、进程推进顺序不当

备注：死锁的4大必要条件是出现死锁时候的状况（现象），  
这里讨论的是产生死锁的根本原因。



## 死锁习题2

- 某系统中只有11台打印机，N个进程共享打印机，每个进程要求3台，当N取值不超过（）时，系统不会发生死锁？
- 最坏情况下，N个进程每个都得到2台打印机，都去申请第3台，为了保证不死锁，此时打印机的剩余数目至少为1台，则：
  - $11 - 2N \geq 1$
  - $N \leq 5$



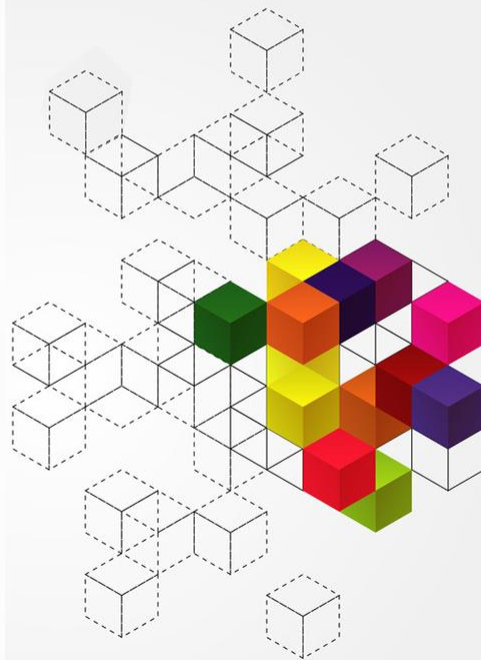
# 死锁习题3

- 设系统中仅有一个资源类，其中共有3个资源实例，使用此类资源的进程共有3个，每个进程至少请求一个资源，它们所需资源最大量的总和为X，则发生死锁的必要条件是（X的取值）：（？）
- 假设3个进程所需该类资源数分别是a,b,c个，因此有：
  - $a+b+c = X$
- 假设发生了死锁，也即当每个进程都申请了部分资源，还需最后一个资源，而此时系统中已经没有了剩余资源，即：
  - $(a-1)+(b-1)+(c-1) \geq 3$
  - $X = a+b+c \geq 6$
- 因此，如果发生死锁，则必须满足的必要条件是  $(X \geq 6)$



# 死锁避免

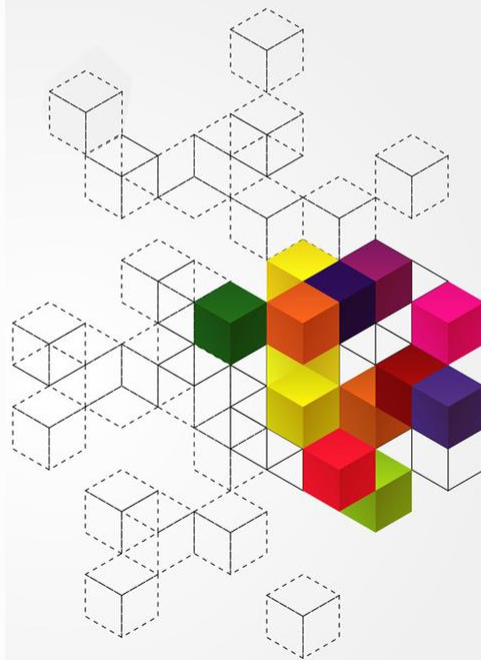
- **银行家算法**：多资源实例条件下的死锁避免算法
- （课件上已有详细说明，此处略过）



# 操作系统Part 3 进程管理

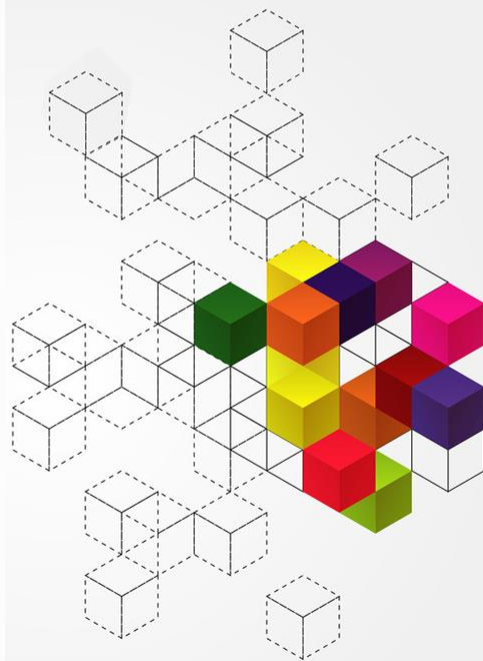
## 处理机调度

- 调度要解决的问题
- 衡量调度策略的指标
- 进程调度的功能
- 调度算法.

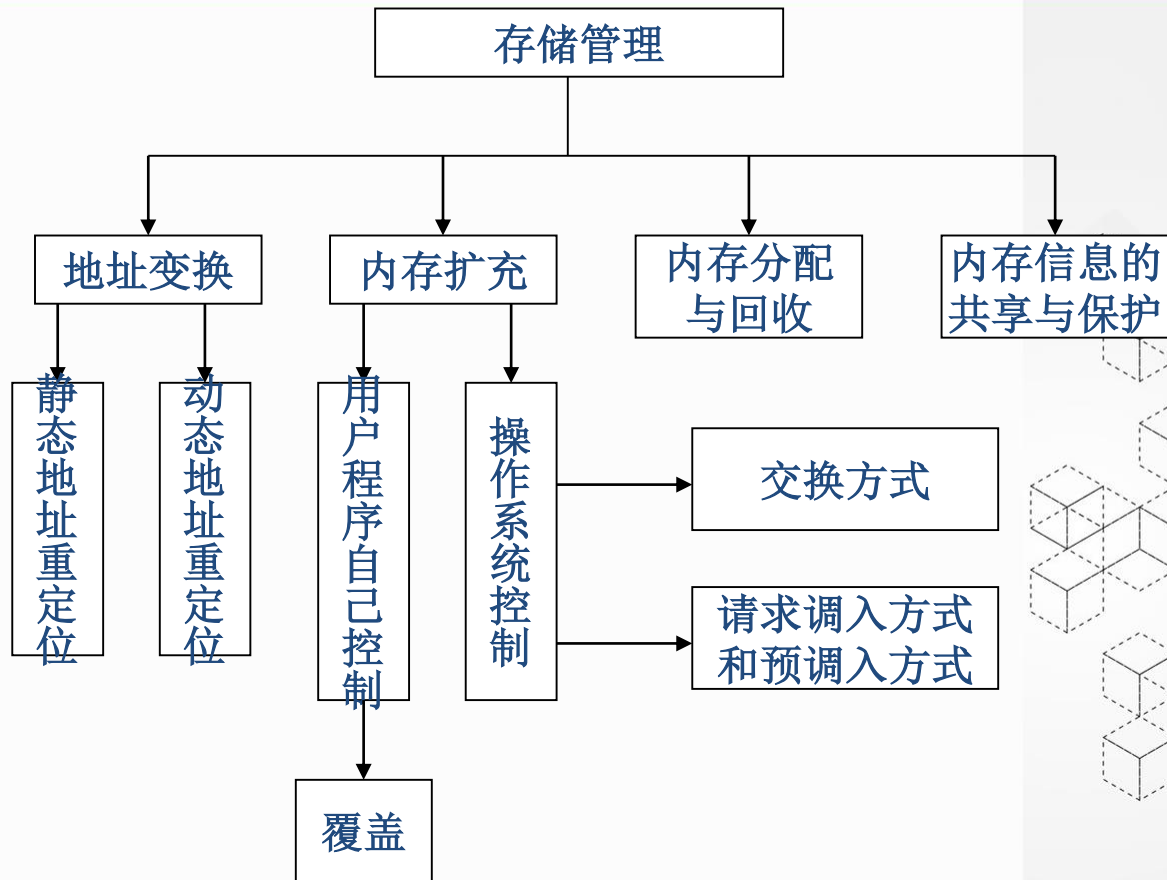


# 操作系统Part 4 存储管理

- 存储管理的功能
- 虚拟存储器概念
- 地址变换方法及各自优缺点
- 内存信息的保护方法
- 存储管理方法和各自的优缺点
- 抖动
- 置换算法
- 工作集



# 内存管理概念结构

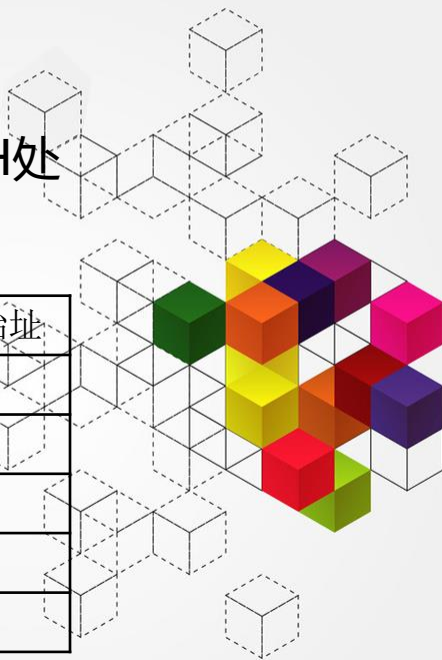




# 存储管理练习

- 某请求页式存储管理中，允许用户编程空间为32个页面（每页2KB），主存为32KB，假定一个用户程序有5页长，且某时刻该用户页面映射如下表。
- (1) 试述有效位、修改位的物理意义。
- (2) 如果分别对以下3个虚地址：0AC5H、1AC5H、3AC5H处进行访问，试计算并说明存储管理系统将如何处理。

虚页号	有效位	访问位	修改位	页面号	外存始址
0	1	0	0	8	5000H
1	1	1	1	4	5800H
2	1	1	0	7	6000H
3	0	0	0	-	6800H
4	0	0	0	-	7000H

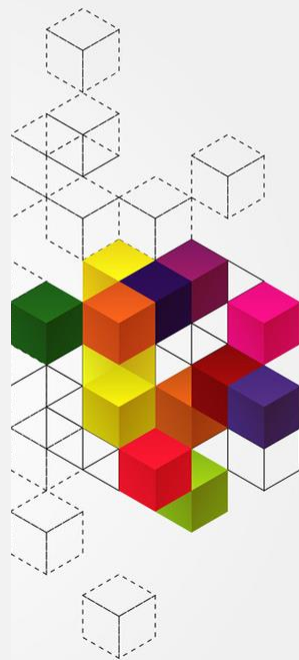


- 答:

- (1) **有效位**表征该页是否在实际内存中，如果在，则状态位为1，反之为0；**修改位**表征该页被调入内存后是否被修改，如果修改，为1，则通过置换算法被换出时，需要写到外存中；反之，通过置换算法被换出时，不需要重新写到外存中。



- 答:
- (2)页面大小为2KB，在虚地址中占有11个二进制位，用户地址空间有32页，虚页号占5位，因此虚地址长度为16位。又因为主存大小为32KB，因此实际物理地址为15位。
- **0AC5H**的二进制形式为 **0000** 1010 1100 0101，其中虚页号为**00001**，后面部分为页内偏移，00001对应十进制的1，由上表可知对应实际页面号为4，化为2进制为 0100，所以应访问的实际物理地址是 0100 010 1100 0101，对应于16进制是 22C5H。
- **1AC5H**的二进制形式为 **0001** 1010 1100 0101，其中虚页号为**00011**，后面部分为页内偏移，00011对应十进制的3，由上表可知虚页号为3的页面没有在内存中，因此发生缺页异常，系统从外存中把第6页调入内存，然后更新页表。
- **3AC5H**的二进制形式为 **0011** 1010 1100 0101，其中虚页号为**00111**，后面部分为页内偏移，00111对应十进制的7，超过了进程的地址空间长度，系统发生地址越界中断，程序运行终止。



## 什么是虚拟存储器?其特点是什么?

- 由进程中的目标代码、数据等的虚拟地址组成的虚拟空间称为**虚拟存储器**。虚拟存储器不考虑物理存储器的大小和信息存放的实际位置，只规定每个进程中相互关联信息的**相对位置**。每个进程都拥有自己的虚拟存储器，且**虚拟存储器的容量是由计算机的地址结构和寻址方式来确定**。
- 实现虚拟存储器要求有相应的地址转换机构，以便把指令的虚拟地址变换为实际物理地址；另外，由于内存空间较小，进程只有部分内容存放于内存中，待执行时根据需要再调指令入内存。

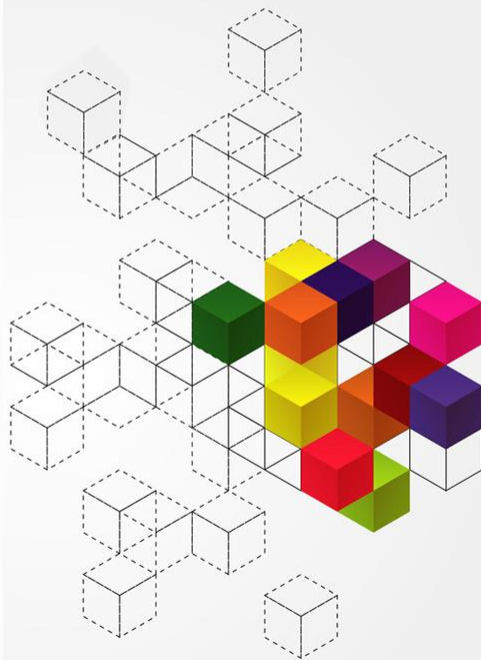


- 实现地址重定位的方法有两种：**静态地址重定位**和**动态地址重定位**。
- (1)**静态地址重定位**是在**虚空间程序执行之前**由**装配程序完成地址映射工作**。静态重定位的优点是不需要硬件支持，但是用静态地址重定位方法，程序经地址重定位后就不能移动了，因而不能重新分配内存，不利于内存的有效利用。静态重定位的另一个缺点是必须占用连续的内存空间和难以做到程序和数据的共享。
- (2)**动态地址重定位**是在程序执行过程中，CPU访问内存之前由硬件地址变换机构将要访问的程序或数据地址转换成内存地址。动态地址重定位的主要优点有：①可以对内存进行非连续分配。②动态重定位提供了实现虚拟存储器的基础。③动态重定位有利于程序段的共享。



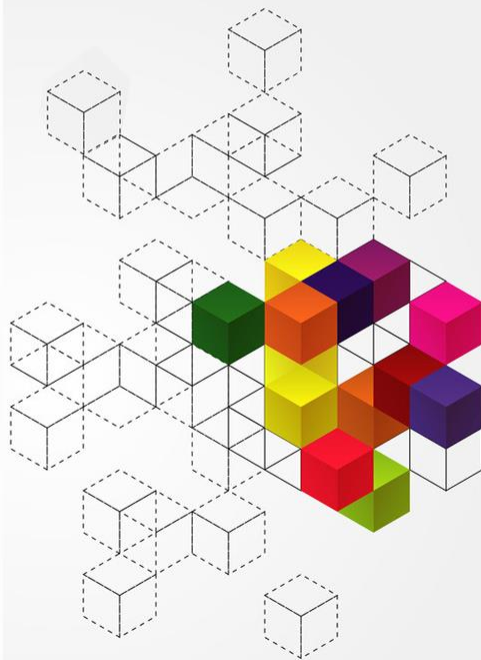
# Part 6 文件系统

- 文件系统的概念和功能
- 文件的访问方式。
- 目录结构各自优缺点
- 文件系统的层次模型
- 虚拟文件系统
- 目录的实现方法及各自优缺点。
- 空闲空间的管理方法



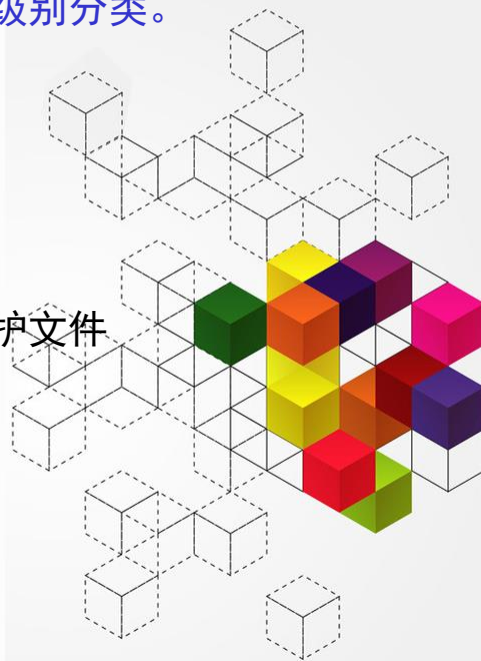
## 8.2 文件系统一般按什么分类？可分为哪几类？

- 文件系统一般按性质、用途、组织形式、文件中的信息流向或文件的保护级别分类。
- 按文件的性质和用途分，分为系统文件、库文件和用户文件。
- 按文件的组织形式分，分为普通文件、目录文件和特殊文件。
- 按文件中的信息流向分，分为输入文件、输出文件、输入输出文件。
- 按文件的保护级别分，分为只读文件、读写文件、可执行文件和不保护文件。



# 8.1 文件一般按什么分类？可分为哪几类？

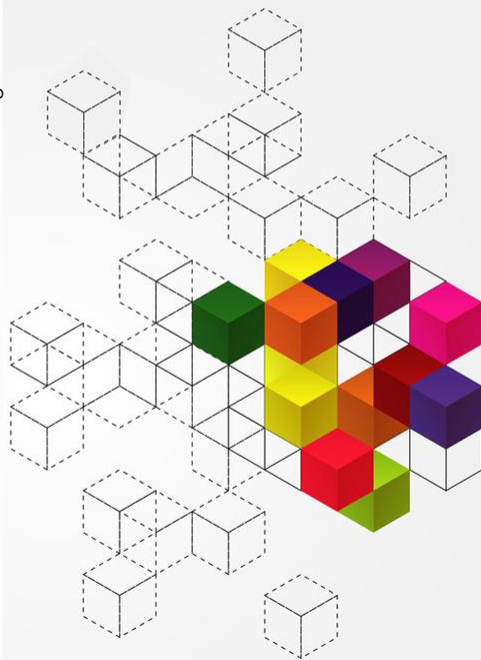
- 文件一般按性质、用途、组织形式、文件中的信息流向或文件的保护级别分类。
- 按文件的性质和用途分，分为系统文件、库文件和用户文件
- 按文件的组织形式分，分为普通文件、目录文件和特殊文件
- 按文件中的信息流向分，分为输入文件、输出文件、输入输出文件
- 按文件的保护级别分，分为只读文件、读写文件、可执行文件和不保护文件





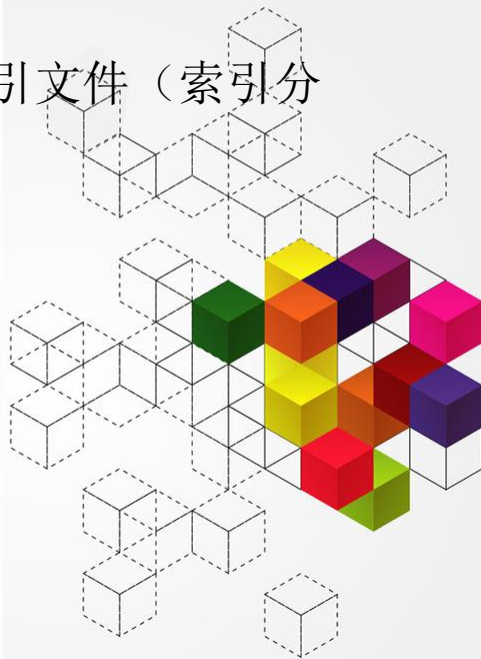
## 8.2 什么是文件的逻辑结构？

- 文件的逻辑结构是用户可见的结构。
- 分为字符流式的无结构文件、简单记录式结构、复杂结构。



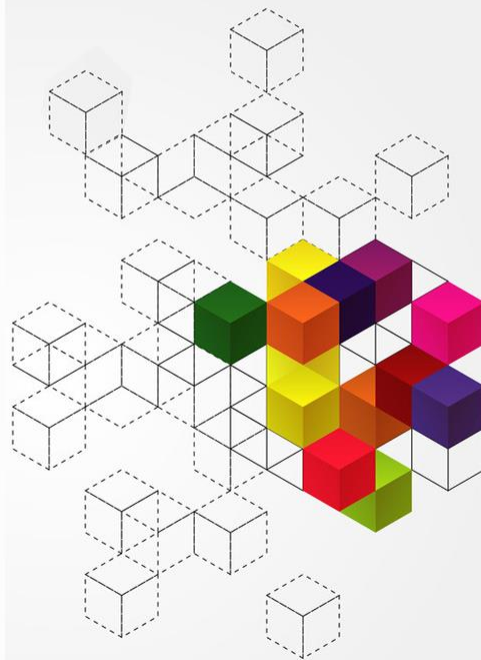
## 8.3 什么是文件的物理结构？

- 文件的物理结构是操作系统内核可见的文件内容结构。
- 分为连续文件（连续分配）、链接文件（链式分配）、索引文件（索引分配）。



# Part 7 设备管理

- 设备分类
- 设备管理的功能和任务
- 数据传送控制方式
- 缓冲技术
- 设备分配所用数据结构及分配过程,分配策略
- 设备驱动程序.



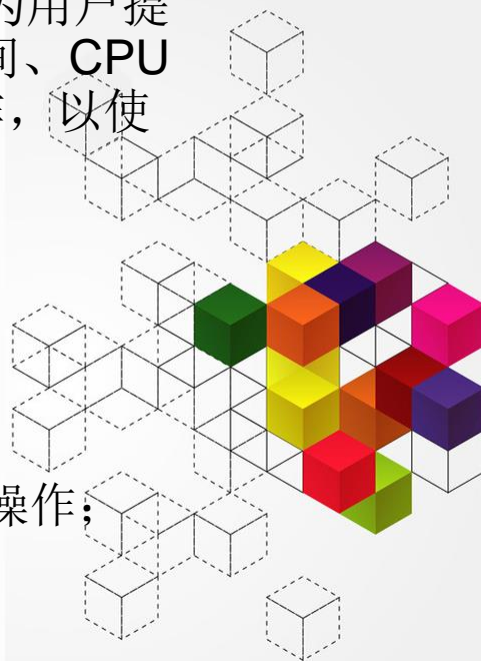
## 设备管理（I/O子系统）的目标和功能是什么？

• 设备管理的目标是

- 选择和分配I/O设备以便进行数据传输操作；
- 控制I/O设备 和CPU(或内存)之间交换数据，为用户提供友好的透明接口，提高设备 和设备之间、CPU和设备之间，以及进程和进程之间的并行操作，以使操作系统获得最佳效率。

设备管理的功能包括：

- (1) 提供和进程管理模块的接口（驱动程序）；
- (2) 进行设备分配；
- (3) 实现设备和设备、设备和CPU 等之间的并行操作；
- (4) 进行缓冲区管理。



## 常用数据传送控制方式有哪几种？试比较它们各自的优缺点。

- 常用数据传送控制方式有程序直接控制方式、中断控制方式、DMA方式。
- 程序直接控制方式就是由用户进程来直接控制内存或CPU和外围设备之间的数据传送。它的优点是控制简单，也不需要多少硬件支持。它的缺点是CPU和外围设备只能串行工作；设备之间只能串行工作，无法发现和处理由于设备或其他硬件所产生的错误。



- **中断控制方式**是利用向CPU发送中断的方式控制外围设备和CPU之间的数据传送。它的优点是大大 **提高了CPU的利用率且能支持多道程序和设备的并行操作**。它的缺点是由于数据缓冲寄存器比较小，如果**中断次数较多**，仍然占用了大量CPU时间；在外围设备较多时，由于中断次数的急剧增加，可能造成CPU无法响应中断而出现**中断丢失**的现象；如果外围设备速度比较快，可能会出现CPU来不及从数据缓冲寄存器中取走数据而丢失数据的情况。
- **DMA方式**是在外围设备和内存之间开辟直接的数据交换通路进行数据传送。它的优点是**除了在数据块传送开始时需要CPU的启动指令，在整个数据块传送结束时需要发中断通知CPU进行中断处理之外，不需要CPU的频繁干涉**。它的缺点是在外围设备越来越多的情况下，多个DMA控制器的同时使用，会引起内存地址的冲突并使得控制过程进一步复杂化。



# I/O设备分类

传输速率

信息交换单位

设备共享特性

分类依据

低速设备

中速设备

高速设备

字符设备

块设备

独占设备

共享设备

虚拟设备

类别

键盘

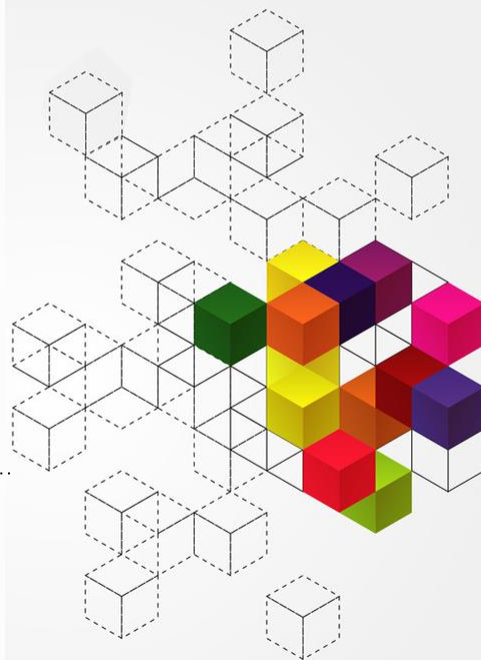
激光打印机

磁盘机

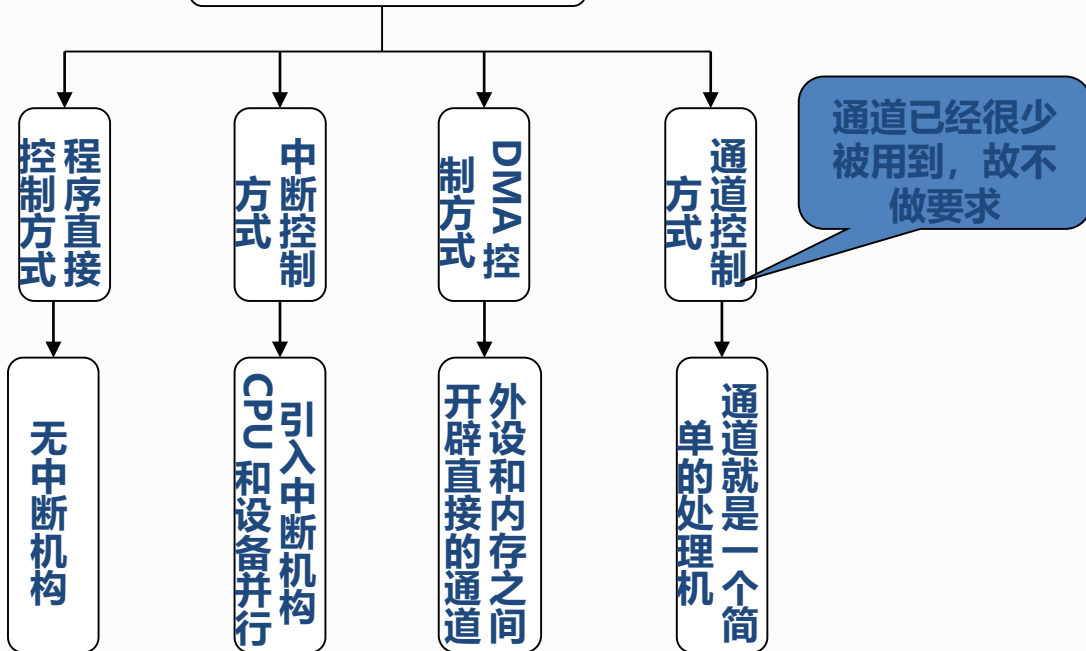
打印机

磁盘

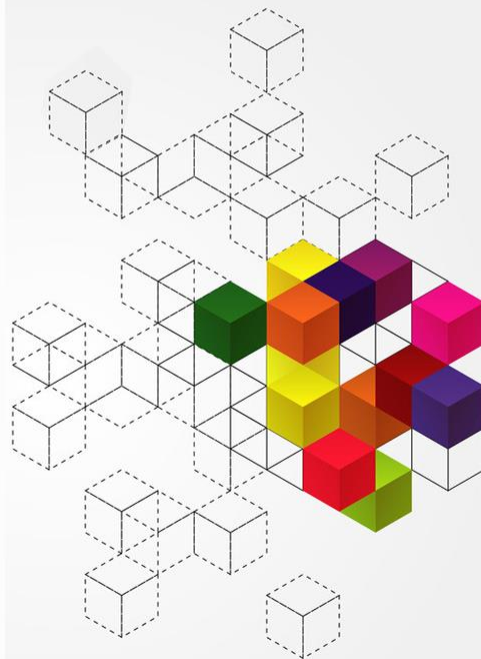
设备类型



## I/O控制方式



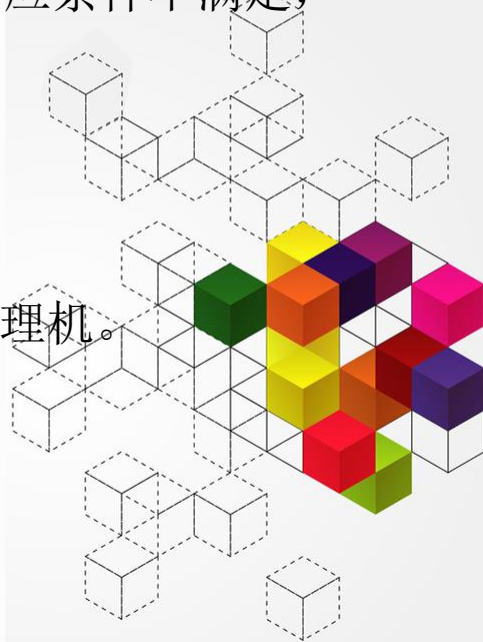
设备管理的主要任务：控制设备和内存或CPU之间的数据传送。



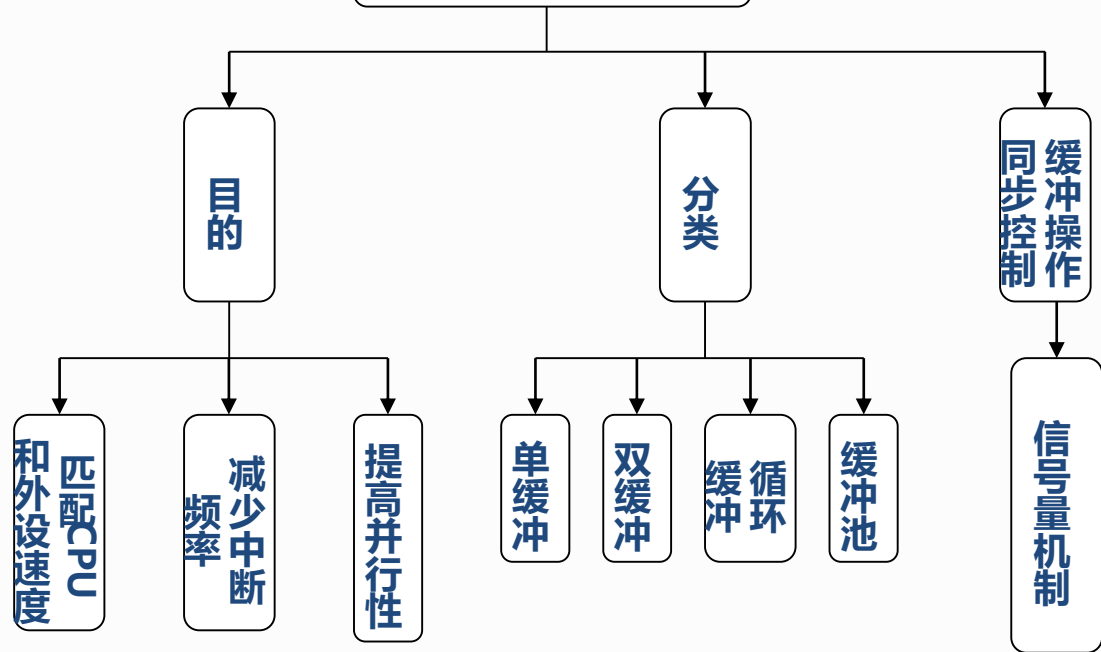


## 描述中断控制方式时的**CPU**动作过程。

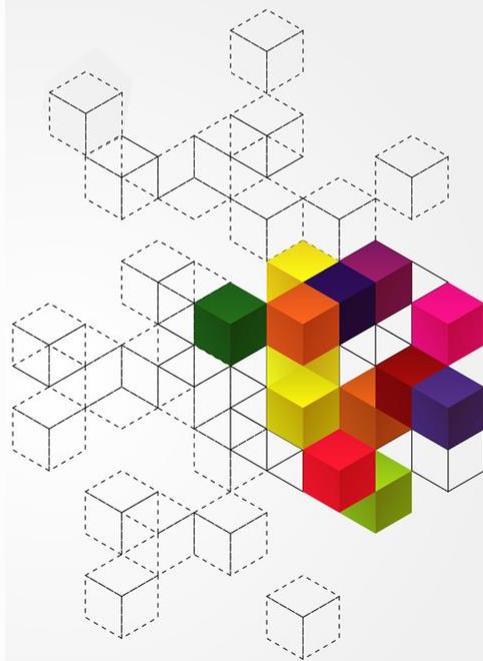
- (1)首先，**CPU**检查响应中断的条件是否满足。如果中断响应条件不满足，则中断处理无法进行。
  - (2)如果**CPU**响应中断，则**CPU**关中断。
  - (3)保存被中断进程现场。
  - (4)分析中断原因，调用中断处理子程序。
  - (5)执行中断处理子程序。
  - (6)退出中断，恢复被中断进程的现场或调度新进程占据处理机。
  - (7)开中断，**CPU**继续执行。



## 缓冲技术



缓冲技术是利用空间来换取时间，加快系统I/O数据处理速度。



## 什么是缓冲?为什么要引入缓冲?

- **缓冲**即是使用专用硬件缓冲器或在内存中一个划出用来暂时存放输入输出数据的区域。
- 引入缓冲是为了匹配外设和**CPU**之间的处理速度，减少中断次数和**CPU**的中断处理时间。

