

本节内容

进程同步 进程互斥

王道考研/CSKAOYAN.COM

知识总览



王道考研/CSKAOYAN.COM

什么是进程同步



知识点回顾：进程具有**异步性**的特征。异步性是指，各并发执行的进程以各自独立的、不可预知的速度向前推进。



有可能这么约
8点~9点：一号
9点~10点：一号
10点~11点：二号
11点~12点：二号



也有可能这么约
8点~9点：一号
9点~10点：二号
10点~11点：一号
11点~12点：二号



一号约会进程的指令1：老渣陪我吃饭
一号约会进程的指令2：老渣把心给我
二号约会进程的指令1：老渣把心给我
二号约会进程的指令2：老渣陪我吃饭



女一号只想做老渣的初恋
女二号只想交一个有恋爱经验的渣男

那么，老渣在并发执行这两个约会进程的时候，就必须保证“一号的指令2”一定要在“二号的指令1”之前执行。



操作系统要提供“**进程同步**机制”来实现上述需求。

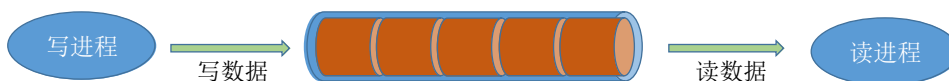
王道考研/CSKAOYAN.COM

什么是进程同步



知识点回顾：进程具有**异步性**的特征。异步性是指，各并发执行的进程以各自独立的、不可预知的速度向前推进。

再看另一个例子：进程通信——管道通信



读进程和写进程并发地运行，由于并发必然导致异步性，因此“写数据”和“读数据”两个操作执行的先后顺序是不确定的。而实际应用中，又必须按照“**写数据**→**读数据**”的顺序来执行的。如何解决这种**异步**问题，就是“**进程同步**”所讨论的内容。

同步亦称**直接制约关系**，它是指为完成某种任务而建立的两个或多个进程，这些进程因为需要在某些位置上**协调**它们的**工作次序**而产生的制约关系。进程间的直接制约关系就是源于它们之间的相互合作。

王道考研/CSKAOYAN.COM

什么是进程互斥



进程的“并发”需要“共享”的支持。各个并发执行的进程不可避免的需要共享一些系统资源（比如内存，又比如打印机、摄像头这样的I/O设备）

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

我们把一个时间段内只允许一个进程使用的资源称为**临界资源**。许多物理设备（比如摄像头、打印机）都属于临界资源。此外还有许多变量、数据、内存缓冲区等都属于临界资源。

对临界资源的访问，必须**互斥**地进行。互斥，亦称**间接制约关系**。**进程互斥**指当一个进程访问某临界资源时，另一个想要访问该临界资源的进程必须等待。当前访问临界资源的进程访问结束，释放该资源之后，另一个进程才能去访问临界资源。

王道考研/CSKAOYAN.COM

什么是进程互斥

对临界资源的互斥访问，可以在逻辑上分为如下四个部分：

```
do {
    entry section;
    critical section;
    exit section;
    remainder section;
} while(true)
```

负责检查是否可进入临界区，若可进入，则应设置**正在访问临界资源的标志**（可理解为“上锁”），以阻止其他进程同时进入临界区

//进入区
//临界区
//退出区
//剩余区

访问临界资源的那段代码

负责解除**正在访问临界资源的标志**（可理解为“解锁”）

做其他处理

注意：

临界区是进程中**访问临界资源**的代码段。
进入区和**退出区**是**负责实现互斥**的代码段。
临界区也可称为“临界段”。



如果一个进程暂时不能进入临界区，那么该进程是否应该一直占着处理机？该进程有没有可能一直进不了临界区？

王道考研/CSKAOYAN.COM

什么是进程互斥

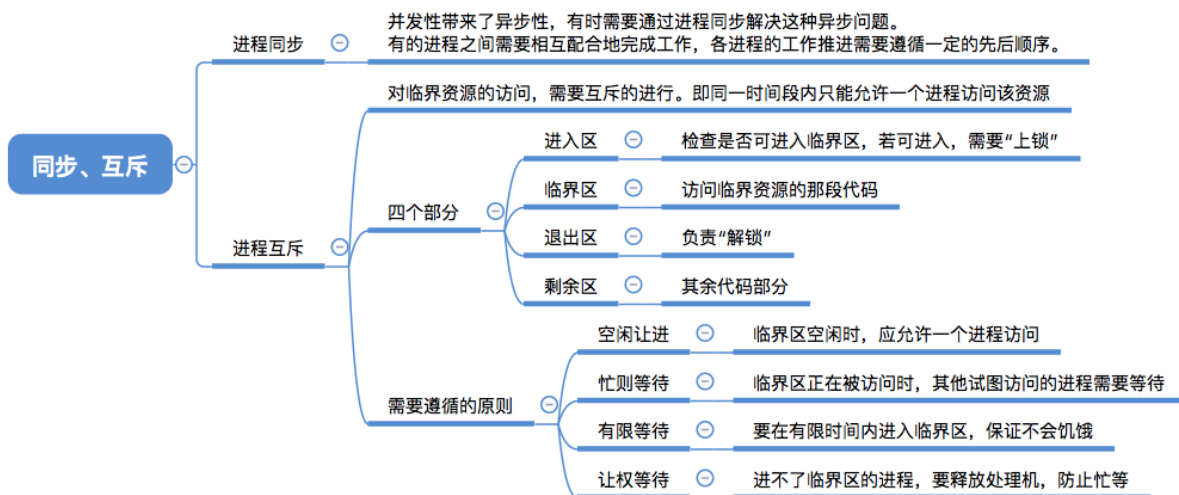
为了实现对临界资源的互斥访问，同时保证系统整体性能，需要遵循以下原则：

1. 空闲让进。临界区空闲时，可以允许一个请求进入临界区的进程立即进入临界区；
2. 忙则等待。当已有进程进入临界区时，其他试图进入临界区的进程必须等待；
3. 有限等待。对请求访问的进程，应保证能在有限时间内进入临界区（保证不会饥饿）；
4. 让权等待。当进程不能进入临界区时，应立即释放处理机，防止进程忙等待。

```
do {
    entry section;    //进入区
    critical section; //临界区
    exit section;     //退出区
    remainder section; //剩余区
} while(true)
```

王道考研/CSKAOYAN.COM

知识回顾与重要考点



王道考研/CSKAOYAN.COM