

# 进程的提出

- 操作系统必须全方位地管理计算机系统中运行的程序
- 因此，操作系统为正在运行程序建立一个管理实体——进程

# 进程的概念

- 进程是一个具有一定独立功能的程序关于某个数据集合的一次运行活动
- 进程是操作系统进行资源分配和调度的一个独立单位

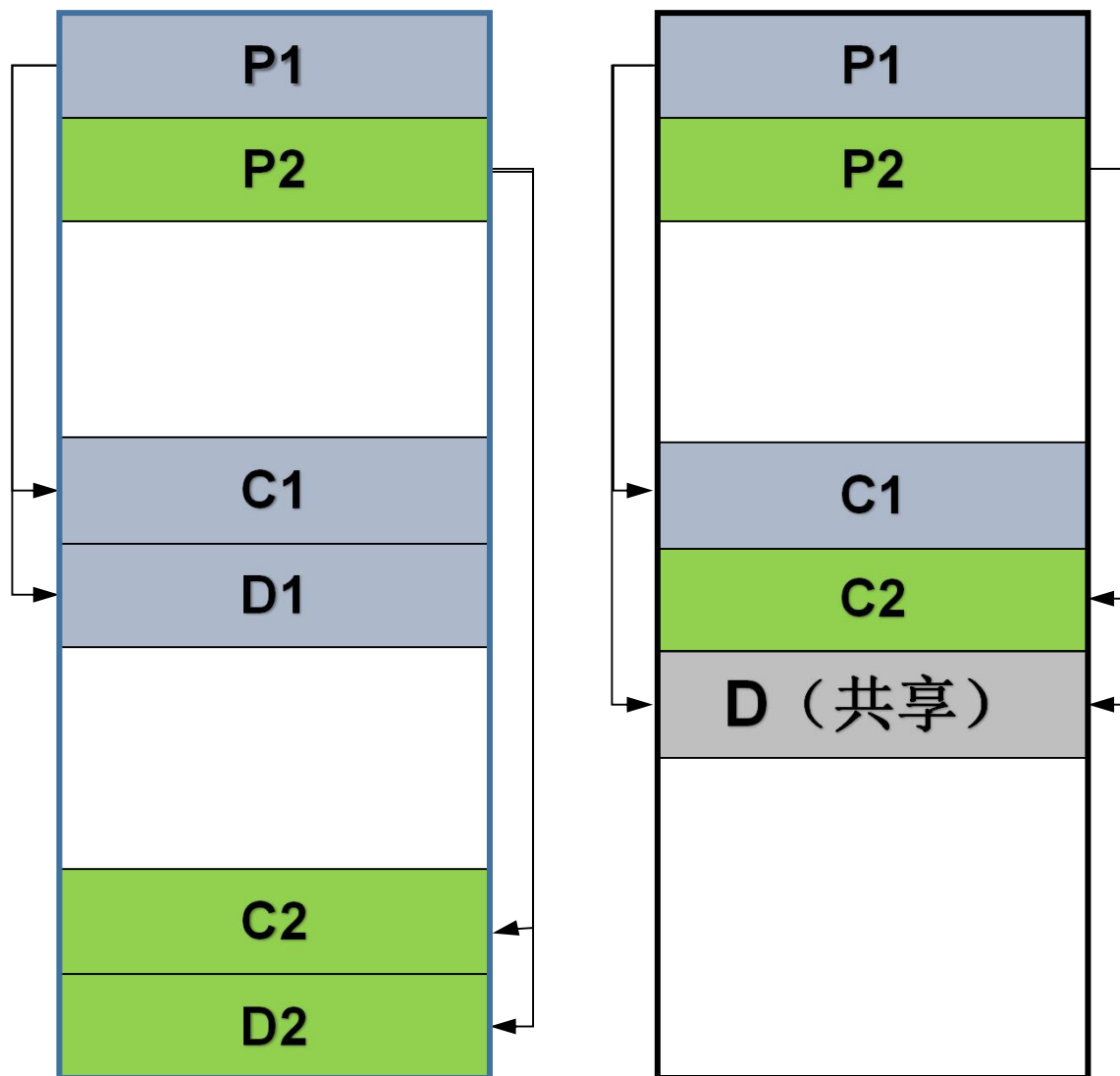
# 进程的概念

- 一个进程包括五个实体部分，分别是：
  - (OS管理运行程序的)数据结构**P**
  - (运行程序的)内存代码**C**
  - (运行程序的)内存数据**D**
  - (运行程序的)通用寄存器信息**R**
  - (OS控制程序执行的)程序状态字信息**PSW**

# 进程举例

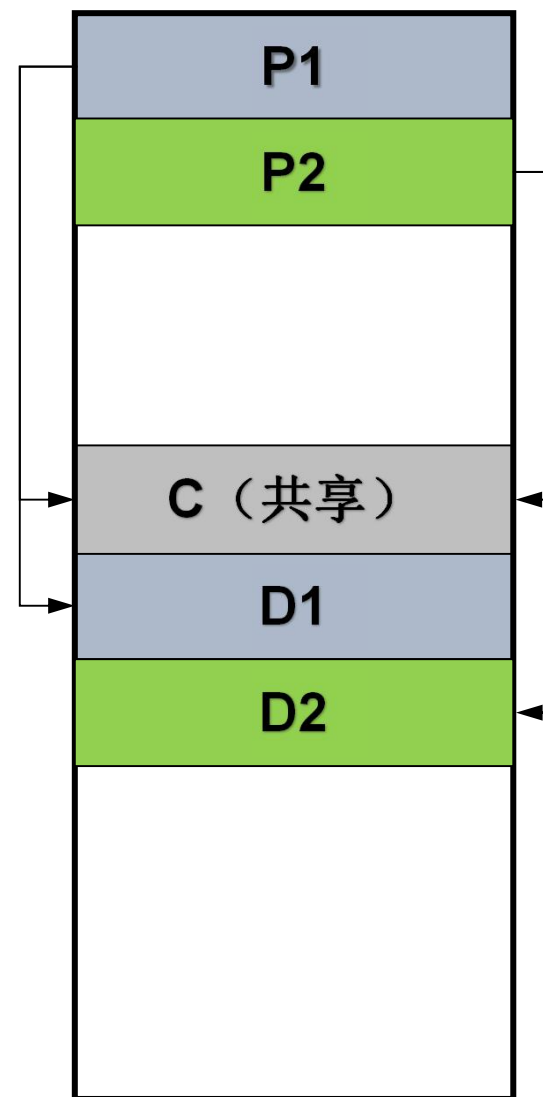
- 不同程序在不同数据集上运行：构成两个无关进程

∞ 不同程序在相同数据集上运行：构成两个共享数据的交往进程



# 进程举例

- 相同代码在不同数据集上运行：构成两个共享代码的无关进程
- 共享的代码称为**可再入程序**，如编辑器
- 可再入程序是**纯代码**的



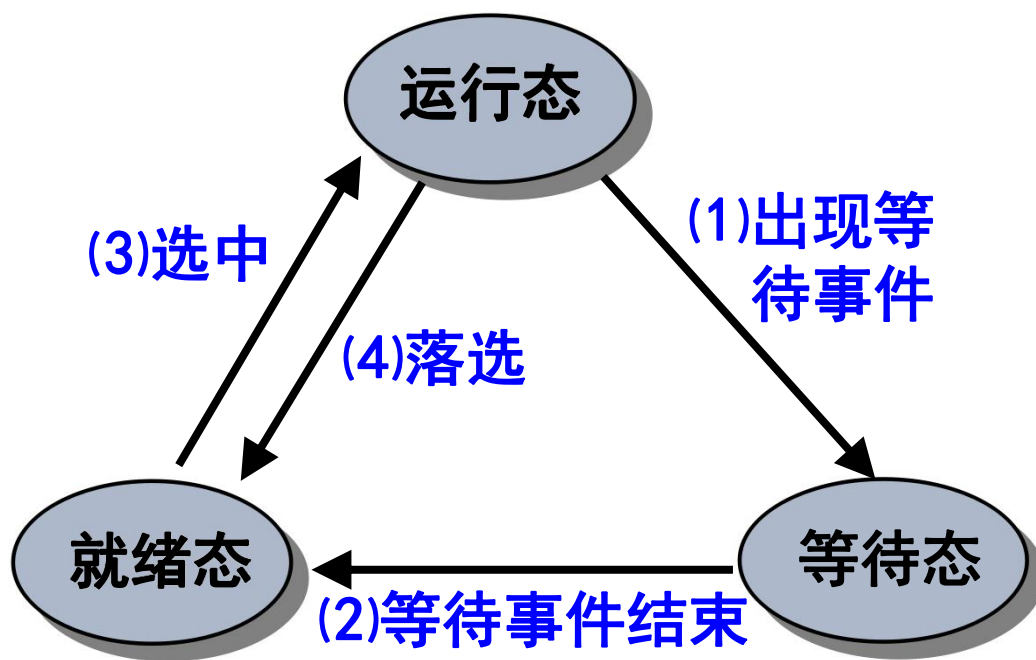
# 进程举例

- 前述的程序与数据集均是内存级的
- 那么，在不同时段中针对(同一个外存数据文件)运行(同一个外存程序文件)，意味着完全不同的（P, C, D, R, Psw）
- 所以两次运行构成两个不同的进程

# 概念级的进程状态

- 运行态指进程占有处理器运行
- 就绪态指进程具备运行条件等待处理器运行
- 等待态指进程由于等待资源、输入输出、信号等而不具备运行条件

# 进程三态模型



(1)运行态→等待态:  
等待资源、I/O、信号

(2)等待态→就绪态:  
资源满足、I/O结束、  
信号完成

(3)就绪态→运行态:  
处理器空闲时选择  
更高优先级进程抢占

(4)运行态→就绪态:  
运行时间片到、  
有更高优先级进程



# 进程挂起的概念

- OS无法预期进程的数目与资源需求，计算机系统在运行过程中可能出现资源不足的情况
- 运行资源不足表现为性能低和死锁两种情况
- 解决办法：剥夺某些进程的内存及其他资源，调入OS管理的对换区，不参加进程调度，待适当时候再调入内存、恢复资源、参与运行
- 这就是进程挂起
- 挂起态与等待态有着本质区别，后者占有已申请到的资源处于等待，前者没有任何资源

# 进程挂起的选择与恢复

- 一般选择等待态进程进入挂起等待态
- 也可选择就绪态进程进入挂起就绪态
- 运行态进程还可以挂起自己
- 等待事件结束后，挂起等待态进入挂起就绪态
- 一般选择挂起就绪态进程予以恢复

