



# 第5章 进程调度



# 目录

- 5.1 基本概念
- 5.2 调度算法
- 5.3 多处理器调度
- 5.4 线程调度
- 5.5 调度实例
- 5.6 算法的评估



## 第5章 进程调度

# 5.1 基本概念

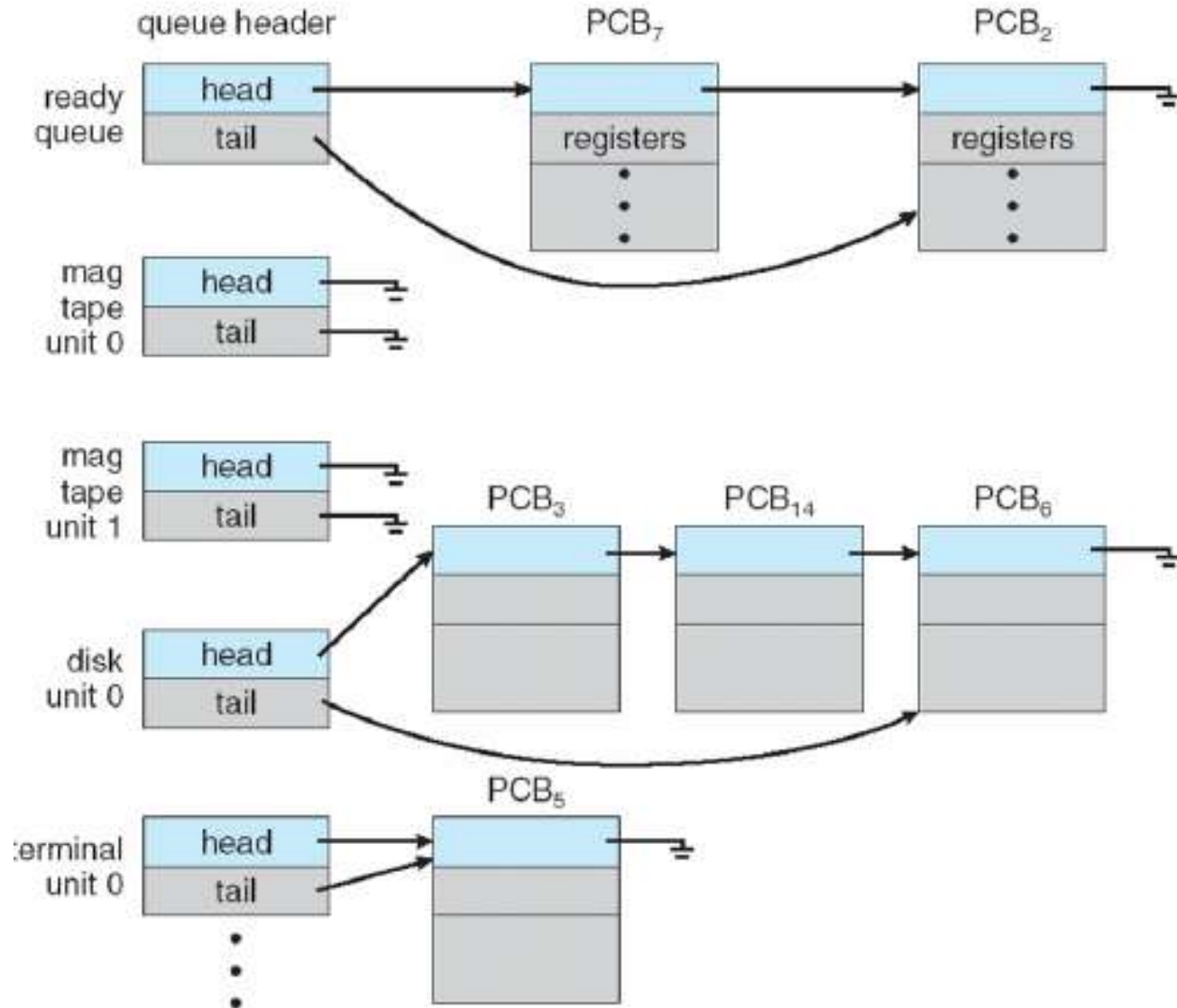


# 1. 调度队列

- 操作系统中维护了一组进程调度队列
  - 作业队列：外存上等待进入内存的作业的集合
  - 就绪队列：所有驻留在内存，并处于就绪状态，等待调度执行，的进程集合
  - 设备队列：等待某个I/O设备的进程集合
  - 进程在整个生命周期中，会在各种调度队列之间迁移



# 调度队列





## 2. 调度的层次

- 一个作业从提交到完成通常要经历多级调度。
  - 高级调度：作业调度
    - 多道批处理系统中，外存上有多个作业等待执行，OS如何决定将哪个作业调入内存？
  - 中级调度：交换调度
    - 内存中的进程越来越多，OS如何缓解内存紧张的问题？
  - 低级调度：进程调度
    - 内存中多个进程竞争CPU，OS如何决定将CPU分配给哪一个进程？



# 调度的层次

- 在不同操作系统中所采用的调度层次不完全相同。
- 有的系统中仅采用一级调度，而在另一些系统中则可能采用两级或三级调度



# 高级调度

## ■ 高级调度：

- 作业调度、宏观调度或长程调度

- 主要任务

- ① 按一定的原则**从外存上**处于后备状态的作业中选择一个或多个作业，**决定哪个进程进入系统，或者说**选择哪些进程可以进入就绪队列。
- ② 为选中作业分配内存、输入/输出设备等必要的资源，并建立相应的进程，以使该作业具有获得竞争处理机的权利。

- 作业调度的运行频率较低，通常为几分钟一次。

- 高级调度决定着多道程序的道数问题





## ■ 作业可分为

- I/O密集型 (I/O-bound) : 执行I/O比计算的时间多, 存在许多短的CPU执行区间 (CPU burst)
- CPU密集型 (CPU-bound) : 执行计算时间多, 有少量的长CPU执行区间

## ■ 作业调度应考虑多类型作业的组合

- 如果I/O密集型过多: 就绪队列为空, 低级调度没什么事儿做
- 如果CPU密集型过多: I/O队列为空, 设备没有得到充分利用
- 因此系统为了保证资源充分利用, I/O密集型与CPU密集型作业应该混合搭配



# 低级调度

## ■ 低级调度

- 进程调度、微观调度或短程调度
- 主要任务
  - ① 按照某种原则决定就绪队列中的哪个进程/内核级线程能获得处理器, 并将处理机出让给它进行工作。
- 进程调度程序是操作系统最为核心的部分, 进程调度策略的优劣直接影响到整个系统的性能。
- 包括: 剥夺式/非剥夺式 (抢占/非抢占)
- 进程调度的运行频率很高, 一般几十毫秒要运行一次。



# 中级调度

## ■ 中级调度

- 负载均衡调度，中程调度，交换调度

- 主要任务

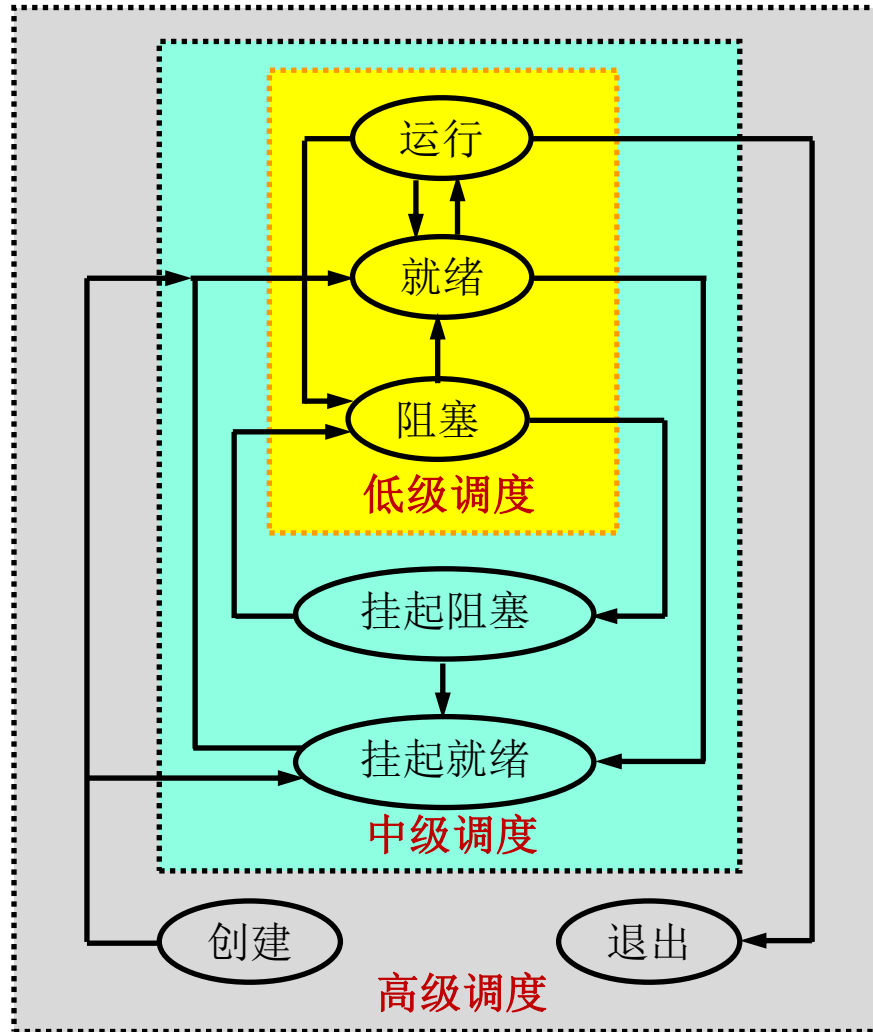
- ① 将内存中暂时不用的信息移到外存，以腾出空间给内存中的进程使用，
- ② 将需要的信息从外存读入内存。

- 中级调度决定了**主存储器中所能容纳的进程数**，这些进程将允许参与竞争处理器资源，提高**内存利用率和系统吞吐量**

- 中级调度根据存储资源量和进程的当前状态来决定辅存和主存中进程的对换，运行频率介于两种调度之间。



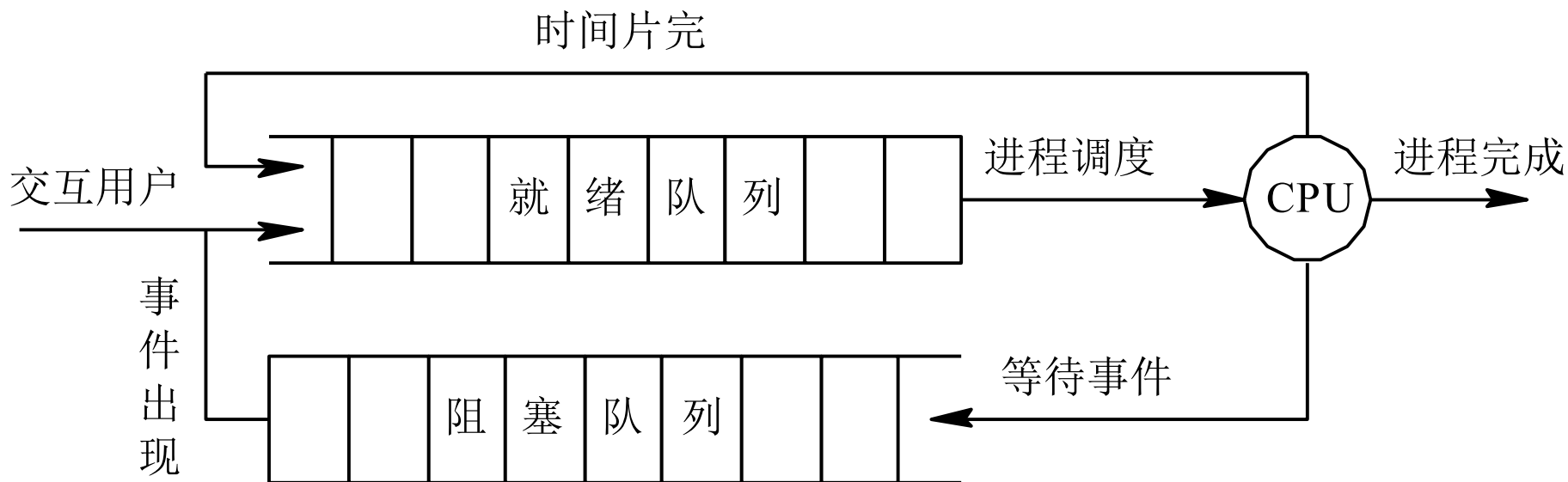
# 调度的层次





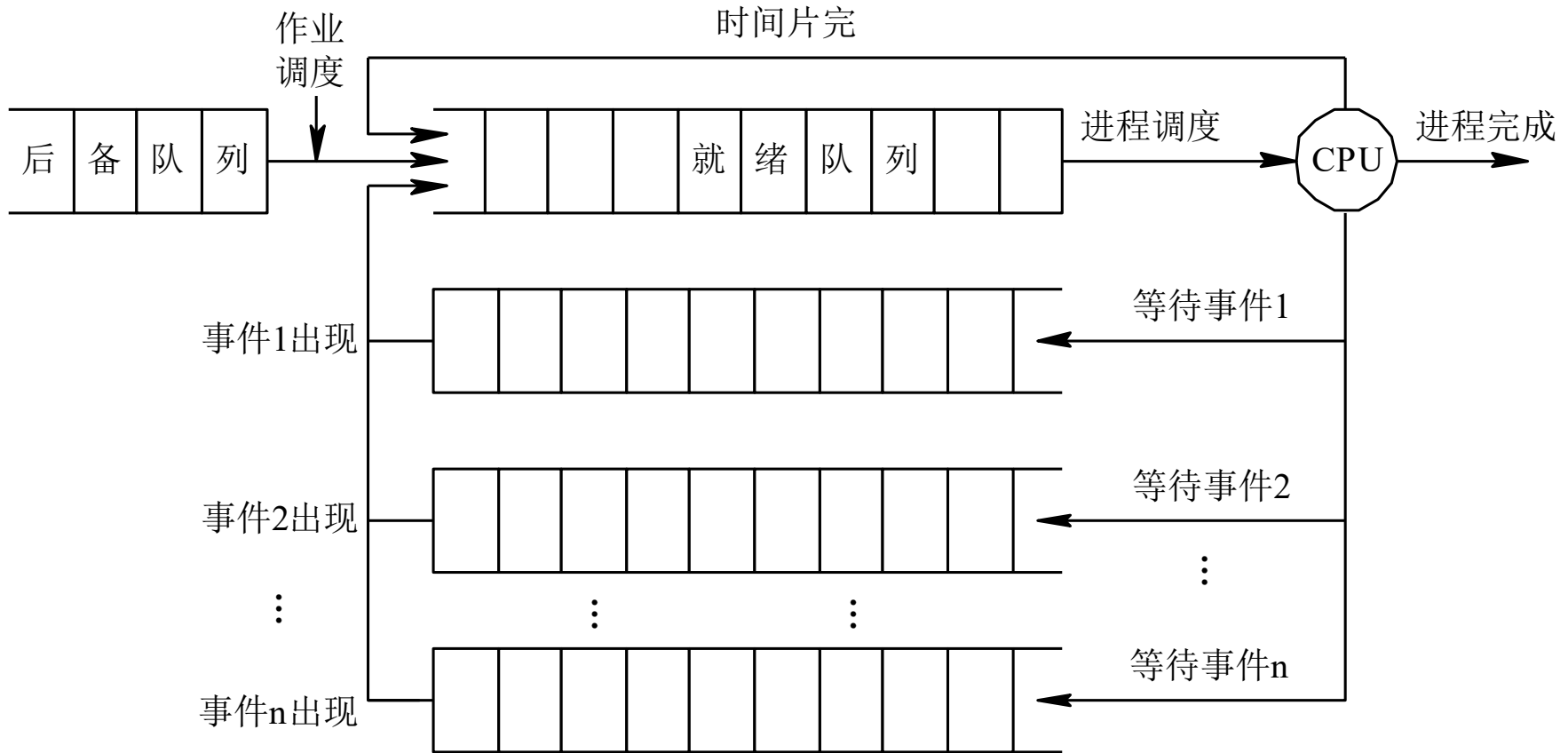
# 3. 调度队列模型

## (1) 仅有进程调度的调度队列模型



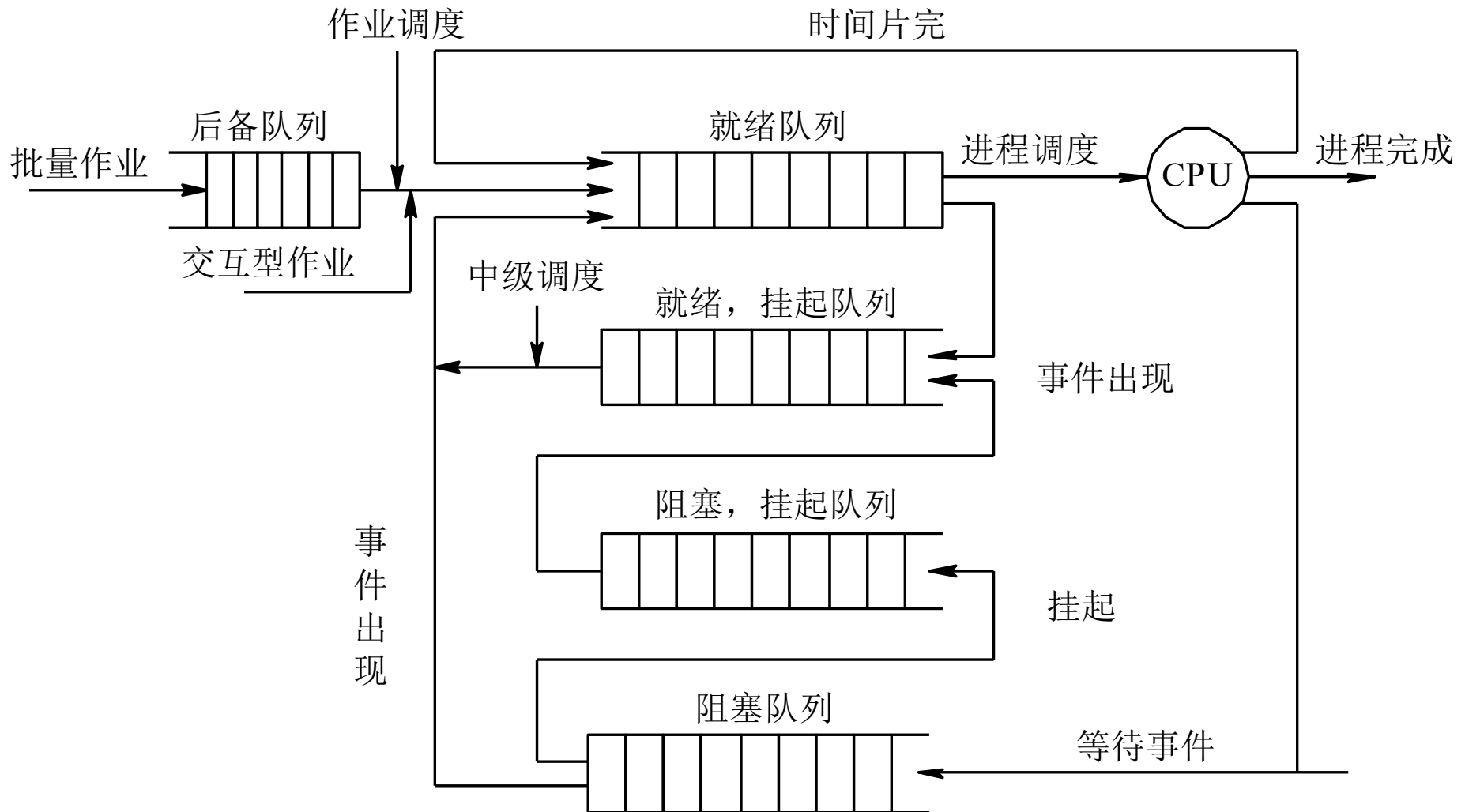


## (2) 具有高、低两级调度的调度队列模型





### (3) 同时具有三级调度的调度队列模型





## 4. 进程调度的有关概念

进程调度要解决的三个问题：

- WHAT：按什么原则选择下一个要执行的进程
  - — 调度算法（调度算法的优劣影响整个系统的性能）
- WHEN：何时选择
  - — 调度时机（抢占式和非抢占式不同）
- HOW：如何让被选中的进程上CPU运行
  - — 调度过程（进程的上下文切换）





# 调度器 (Scheduler)

- 进程调度程序主要完成以下功能：
  - 记录系统中所有进程的状态、优先数和资源情况。
  - 依据调度算法，从就绪队列中选择获得处理机的进程。
  - 实施处理机的分配及回收。



# 分派器(Dispatcher)

- 分派器模块负责将CPU的控制交给调度器所选定的进程，其任务包括：
  - 切换上下文
  - 切换用户模式
  - 跳转到用户程序的合适位置，并重启程序
- 分派延迟
  - 分派器从停止一个进程到启动另一个进程所花的时间。

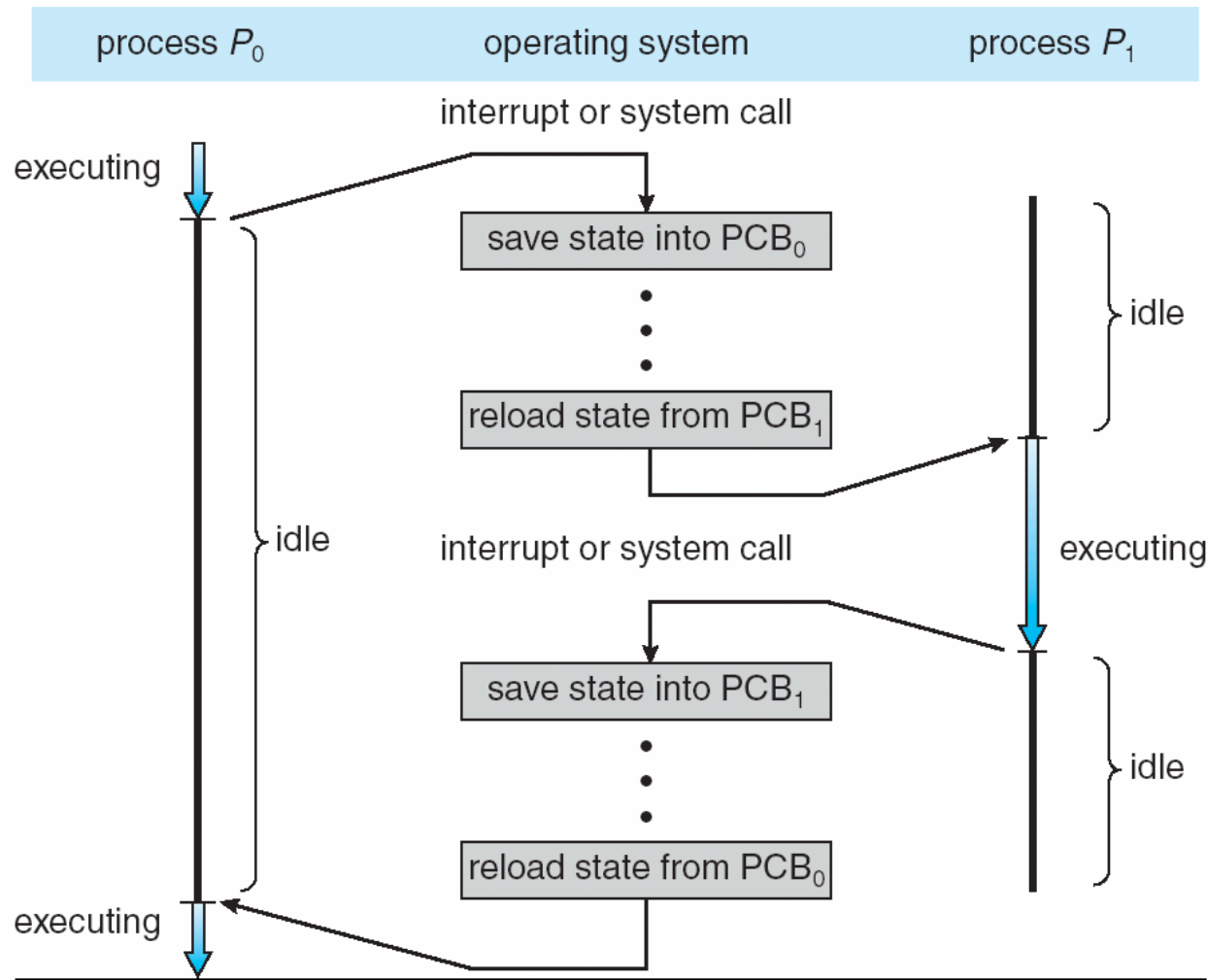


# 上下文切换

- 将CPU切换到另一个进程需要保存当前进程的状态并恢复另一个进程的状态，这一任务称为上下文切换。
- 上下文信息保存到进程的PCB中。
- 上下文切换时间是系统开销，与硬件密切相关。



# 在进程切换时保存上下文信息到PCB



进程切换和CPU模式切换是怎样的？



# 进程调度方式

- 进程调度有两种方式：
  - 非抢占方式
  - 抢占方式



# 非抢占方式 (Nonpreemptive)

- **非抢占方式**：又称非剥夺方式、不可剥夺方式、不可抢占方式。这种调度方式是指一旦将处理机分配给某进程后，便让该进程**一直执行**，直到该进程完成或发生某事件而进入阻塞状态，才把处理机分配给其他进程。
- **特点**：简单，系统开销小，但无法处理紧急任务。



# 抢占方式 (Preemptive)

- **抢占方式**：又称剥夺方式、可剥夺方式。这种调度方式是指允许调度程序根据某种原则去停止正在执行的进程，将已分配给该进程的处理机重新分配给其他进程。
- **抢占原则**有：  
    优先权、短作业优先、时间片等。
- **特点**：系统响应性高，但增加系统开销



# 进程调度发生的时机

## ■ 非抢占方式

- 当一个进程终止
- 当一个进程由运行态转到等待态

## ■ 抢占方式

- 当一个进程终止
- 当一个进程由运行态转到等待态
- 当一个进程由运行态转到就绪态
- 当一个进程由等待态转到就绪态





## 5. 调度性能的评价准则

调度的最根本目标：

- 保证CPU资源和其他资源利用的合理性和有效性
- **合理性**：根据调度策略合理分配处理机，既要保证系统实现特殊功能的要求，又要使各个作业合理使用处理机资源
- **有效性**：按调度策略使处理机和I/O设备得到合理有效的分配，资源充分利用



# 调度性能的评价准则(Cont.)

## ① 资源利用率

- CPU利用率 = CPU有效工作时间 / CPU总的运行时间,
- CPU总的运行时间 = CPU有效工作时间 + CPU空闲等待时间。
- $\text{CPU利用率} = 1 / (1 + \text{CPU空闲等待时间} / \text{CPU有效工作时间})$
- 真实系统介于40%~90%之间



# 调度性能的评价准则(Cont.)

## ② 响应时间

- 交互式进程从提交一个请求(命令)到接收到第一个响应之间的时间间隔称响应时间。
- 使交互式用户的响应时间尽可能短，或尽快处理实时任务。
- 这是分时系统和实时系统衡量调度性能的一个重要指标。
- 对于分时系统，最小化响应时间的方差比最小化平均相应时间更为重要。



# 调度性能的评价准则 (Cont.)

## ③ 周转时间

- 从作业（进程）提交给系统开始，到作业（进程）完成为止的时间间隔称周转时间，
- 作业周转时间是衡量批处理系统的一个重要指标
- 应使作业周转时间或平均作业周转时间尽可能短。



# 周转时间与平均周转时间

## ■ 周转时间

- 作业的**周转时间**是指从作业提交到作业完成之间的时间间隔。周转时间是所有时间段之和，包括等待进入内存、在就绪队列中等待、在CPU上执行和I/O执行的时间。这段时间间隔包括了，作业等待、挂起等。

- $T_i = T_{e_i} - T_{s_i}$

## ■ 平均周转时间

- 是指多个作业的周转时间的平均值。n个作业的平均周转时间：

- $T = (T_1 + T_2 + \dots + T_n) / n$  ( $T_i$ 为作业 i 的周转时间)



# 带权周转时间与平均带权周转时间

## ■ 带权周转时间

- **带权周转时间**是指作业周转时间与作业实际运行时间的**比值**， $W_i$ 。
- **注意**：这里**作业实际运行时间**是指在CPU中的时间，不包括阻塞、挂起等时间

## ■ 平均带权周转时间

- 是指多个作业的带权周转时间的平均值。n 个作业的平均带权周转时间：
- $W = (W_1 + W_2 + \dots + W_n) / n$  ( $W_i$ 为作业 i 的带权周转时间)



# 调度性能的评价准则(Cont.)

## ④ 等待时间

- 对于进程，等待时间是进程在就绪队列中等待调度的时间总和
- 对于作业，等待时间是作业在后备作业队列和对应进程在就绪队列中等待时间的总和

## ⑤ 系统吞吐率（系统吞吐量）

- 单位时间内CPU完成进程(作业)的数量。
- 长作业可以1个/小时，短作业可以10个/秒



# 调度性能的评价准则(Cont.)

- 最优准则: Optimization Criteria:
  - 最大的CPU利用率                      Max CPU utilization
  - 最大的吞吐量                         Max throughput
  - 最短的周转时间                       Min turnaround time
  - 最短的等待时间                       Min waiting time
  - 最短的响应时间                       Min response time
- 公平性
  - 确保每个用户每个进程获得合理的CPU份额或其他资源份额，不会出现饿死情况。
  - 对于不同的系统设计目标，公平性原则是不同的
    - 批处理系统、分时系统、实时系统





# 讨论

- 思考一下，你的OS存在哪些层次的调度？
- CPU利用率，提到了CPU空闲等待时间，它在等待什么？
- 分析一下，有了周转时间，为什么还要引入平均XX、带权XX、平均带权XX？会有什么用处？



# 附录：作业调度

- **作业**是用户在一次解题或一个事务处理过程中要求计算机系统所做工作的集合，包括用户程序、所需的数据及命令等。
- 计算机系统在完成一个作业的过程中所做的一项相对独立的工作称为一个**作业步**。
- 例如，在编制程序过程中通常要进行编辑输入、编译、链接、运行几个作业步。



# 作业与进程的关系

- 作业是**任务实体**
- 进程是完成任务的**执行实体**
  - 没有作业任务，进程无事可干
  - 没有进程，作业任务没法完成。
- 作业概念更多地用在**批处理操作系统**
- 进程则可以用在各种**多道程序设计系统**

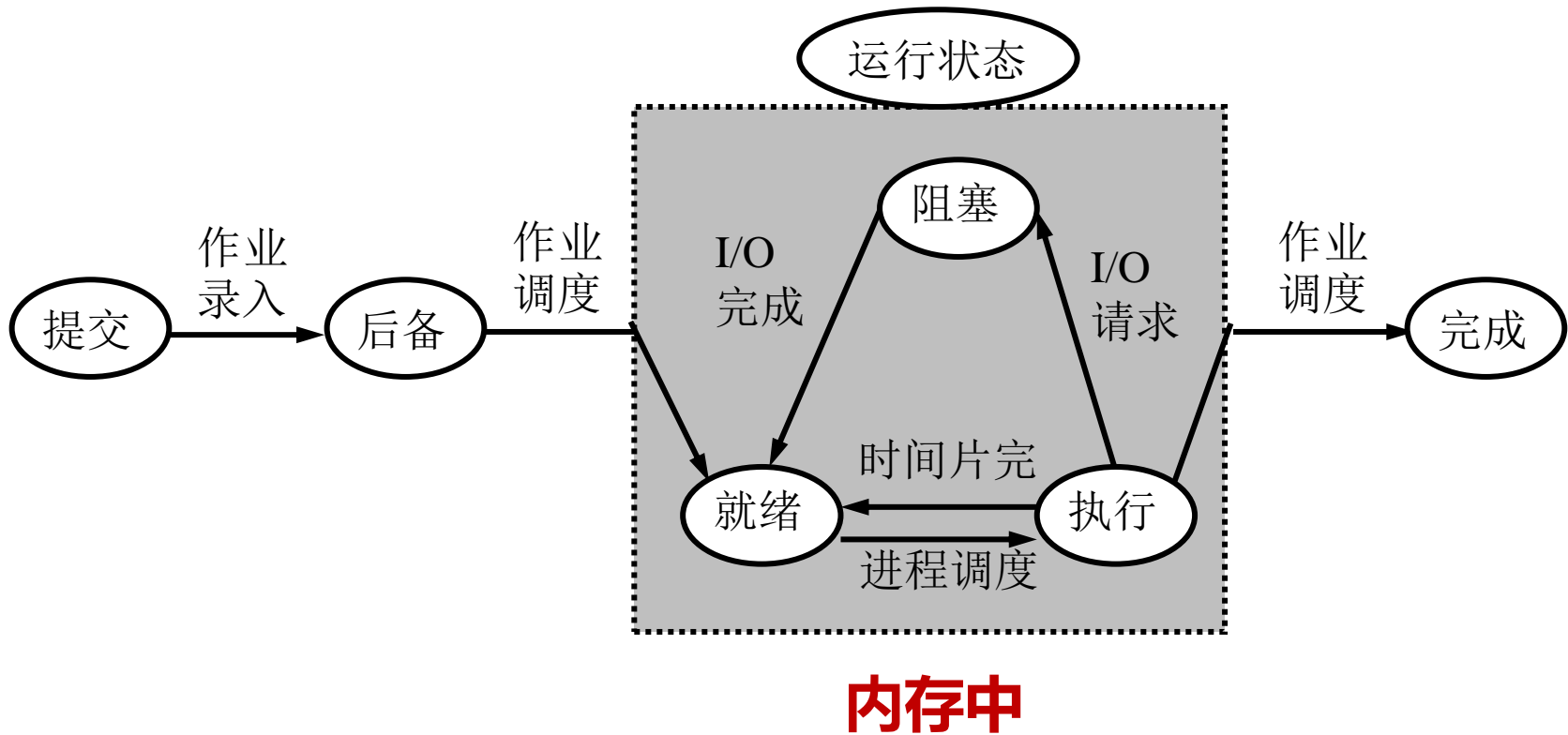


# 作业的状态及转换

- 作业从提交到完成要经历四种状态：
  - ① **提交状态**：用户作业由输入设备，向**系统外存**输入时作业所处的状态。
  - ② **后备状态**：作业输入到外存后，系统为其建立了作业控制块，并把它插入到后备作业队列中等待调度运行。
  - ③ **运行状态**：作业在**内存中**，**包含占有CPU&没占有CPU**
  - ④ **完成状态**：作业正常或异常结束，但作业占有的资源还未被系统全部回收。



# 作业状态转换图





# 作业调度过程

- 作业调度程序主要完成以下工作
  - ① 记录进入系统的各个作业情况。
  - ② 从后备作业中挑选一些作业投入执行。
  - ③ 为被选中的作业做好执行前的准备工作。
  - ④ 在作业运行结束或运行过程中因某种原因需要撤离时，作业调度程序还要完成作业的善后处理工作。



# 作业控制块

- 为管理作业，系统设置了作业控制块。系统通过JCB感知作业的存在，JCB是作业存在的唯一标志。
- 通常作业控制块中包括的主要内容有：
  - 资源要求。
  - 资源使用情况。
  - 作业的控制方式、类型和优先级等。
  - 作业名、作业状态。



# 作业控制块

## ■ 资源要求

- 资源要求是指作业运行需要的资源情况，包括：估计运行时间、最迟完成时间、需要的内存容量、外设类型及数量等。

## ■ 资源使用情况

- 资源使用情况包括作业进入系统的时间、开始运行时间、已运行时间、内存地址、外设台号等。





# 作业控制方式、类型和优先级

- 作业的控制方式有联机作业控制和脱机作业控制。
- 从不同角度出发可以对作业进行不同的分类，如终端型和批量型，
- 作业的优先级是指作业进入系统运行的优先级别，优先级高的作业可以优先进入系统运行。