

第3章 调度

3.2 作业、进程和线程的调度



1 作业调度的概念

- **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

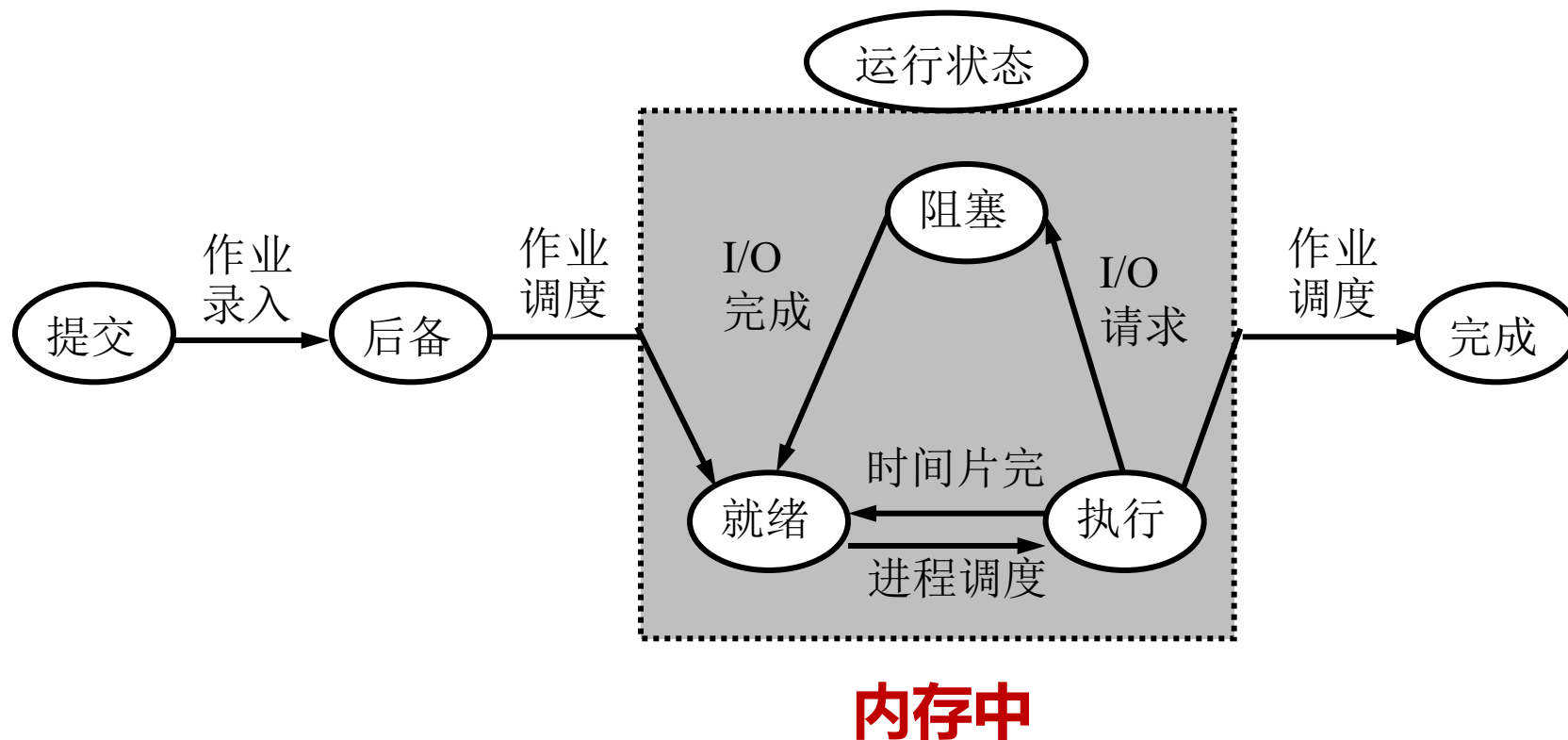
- 作业是**任务实体**
- 进程是完成任务的**执行实体**
 - 没有作业任务，进程无事可干
 - 没有进程，作业任务没法完成。
- 作业概念更多地用在**批处理操作系统**
- 进程则可以用在各种**多道程序设计系统**



2 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





3 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。



4 进程调度概念

- 进程调度程序主要完成以下功能：
 - ① 记录系统中所有进程的状态、优先数和资源情况。
 - ② 从就绪队列中选择获得处理机的进程。
 - ③ 实施处理机的分配及回收。



5 进程调度的方式

- 进程调度/低级调度，有两种方式：
 - 抢占方式/剥夺方式
 - 非抢占方式/非剥夺方式



抢占方式

- **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：高优先级抢占/剥夺低优先级
 - 时间片：时间片用完，剥夺CPU使用权



非抢占方式

- **非抢占方式**：又称非剥夺方式、不可剥夺方式、不可抢占方式。
 - 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。
- 非抢占方式中引起进程调度的因素有：
 - 进程结束
 - 因某种原因而阻塞
 - 执行同步原语等。
- 特点：简单，系统开销小，但无法处理紧急任务。



引起进程调度的原因

当一个进程：

- 从运行态切换到等待状态
 - I/O请求，调用wait等
- 从运行状态切换到就绪状态
 - 如发生中断
- 从等待状态切换到就绪状态
 - 如系统调用或者中断返回
- 进程终止



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。
- 一点回顾：
 - 进程状态切换和CPU模式切换是怎样的？
 - 调度程序和分派器是一气呵成的吗？



6. 线程调度

- 线程调度要区分用户级和内核级线程
- 当OS支持线程时，内核级线程（非用户级线程）是操作系统调度的。
 - Q:为什么不是用户级线程呢？
- 对于多对一和多对多模型的系统库，负责调度用户级线程
 - 此时，竞争CPU发生在**一个进程的不同线程之间**，称为**进程竞争范围PCS**（Process-Contention Scope）
 - **内核级的线程调度**采用，**系统竞争范围SCS**（System-Contention Scope），竞争CPU发生在**系统内所有的线程之间**。



Pthread 调度

- Pthread API可以提供纯线程级或者内核支持的线程库。
- 提供提供线程创建时指定PCS、SCS
 - PTHREAD_SCOPE_PROCESS : PCS
 - PTHREAD_SCOPE_SYSTEM : SCS
- 和具体内核有关



讨论

- ① 作业有抢占/非抢占的概念吗？
- ② 前面我们说的那些软硬件机制，哪些是支持抢占的？如何写程序设计一个抢占过程？
- ③ 引起调度原因中这个进程状态变化的原因，是由OS造成，还是其他进程造成？
- ④ 当只发生以上哪几种调度情况，是非抢占调度？