



# 计算机操作系统

## 2 处理器管理 – 2.3 进程管理

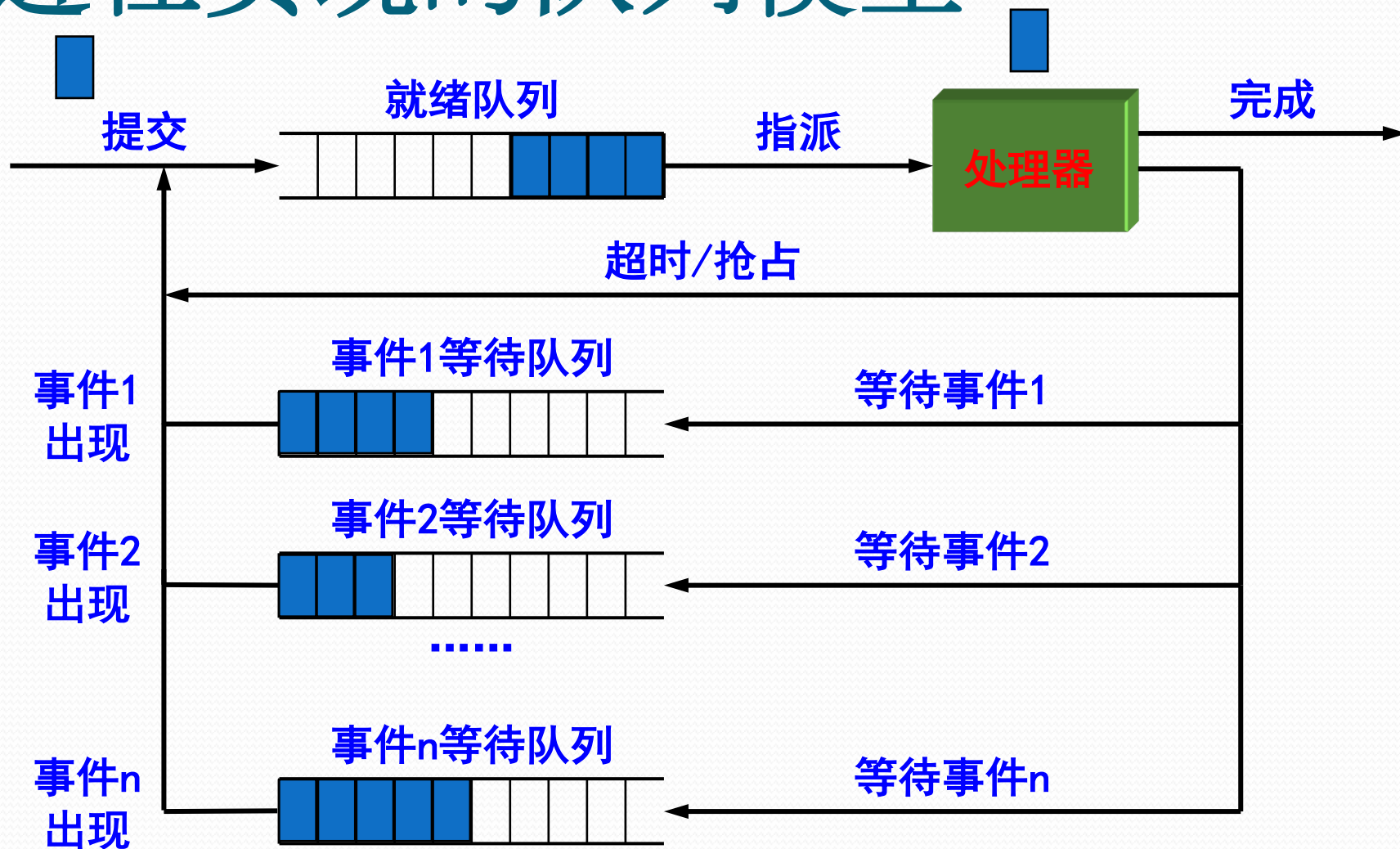
### 2.3.3 进程管理的实现

了解进程管理程序的概念级组成  
掌握进程队列模型和队列管理模块  
理解进程的控制与管理程序的实现  
掌握进程切换、模式切换及其过程

# 概念级的OS进程管理软件

- 关键的进程管理软件包括：
  - 系统调用/中断/异常处理程序
  - 队列管理模块
  - 进程控制程序
  - 进程调度程序（独立进程居多）
  - 进程通信程序（多个程序包）
  - 终端登录与作业控制程序、性能监控程序、审计程序等外围程序

# 进程实现的队列模型



# 队列管理模块

- 队列管理模块是操作系统实现进程管理的核心模块
- 操作系统建立多个进程队列，包括就绪队列和等待队列
- 按需组织为先进先出队列与优先队列
- 队列中的进程可以通过PCB中的队列指引元采用单/双指引元或索引连接
- 出队和入队操作
- 进程与资源调度围绕进程队列展开

# 进程的控制与管理

- 进程创建：进程表加一项，申请PCB并初始化，生成标识，建立映像，分配资源，移入就绪队列
- 进程撤销：从队列中移除，归还资源，撤销标识，回收PCB，移除进程表项
- 进程阻塞：保存现场信息，修改PCB，移入等待队列，调度其他进程执行
- 进程唤醒：等待队列中移出，修改PCB，移入就绪队列（该进程优先级高于运行进程触发抢占）
- 进程挂起：修改状态并出入相关队列，收回内存等资源送至对换区
- 进程激活：分配内存，修改状态并出入相关队列
- 其他：如修改进程特权

# 原语与进程控制原语

- 进程控制过程中涉及对OS核心数据结构(进程表/PCB池/队列/资源表)的修改
- 为防止与时间有关的错误，应使用原语
- 原语是由若干条指令构成的完成某种特定功能的程序，执行上具有不可分割性
- 原语的执行可以通过关中断实现
- 进程控制使用的原语称为进程控制原语
- 另一类常用原语是进程通信原语



# 进程切换

- 进程切换指从正在运行的进程中收回处理器，让待运行进程来占有处理器运行
- 进程切换实质上就是被中断运行进程与待运行进程的上下文切换，处理过程是：
  - 保存被中断进程的上下文
  - 转向进程调度
  - 恢复待运行进程的上下文

# 模式切换

- 进程切换必须在操作系统内核模式下完成，这就需要模式切换
- 模式切换又称处理器状态切换，包括：
  - 用户模式到内核模式  
由中断/异常/系统调用中断用户进程执行而触发
  - 内核模式到用户模式  
OS执行中断返回指令将控制权交还用户进程而触发



# 模式切换的基本工作任务

- 中断装置完成正向模式切换，包括：
  - 处理器模式转为内核模式
  - 保存当前进程的PC/PSW值到核心栈
  - 转向中断/异常/系统调用处理程序
- 中断返回指令完成逆向模式转换，包括：
  - 从待运行进程核心栈中弹出PSW/PC值
  - 处理器模式转为用户模式

# 进程切换的工作过程

1. (中断/异常等触发) 正向模式切换并压入PSW/PC
2. 保存被中断进程的现场信息
3. 处理具体中断/异常
4. 把被中断进程的系统堆栈指针SP值保存到PCB
5. 调整被中断进程的PCB信息, 如进程状态
6. 把被中断进程的PCB加入相关队列
7. 选择下一个占用CPU运行的进程
8. 修改被选中进程的PCB信息, 如进程状态
9. 设置被选中进程的地址空间, 恢复存储管理信息
10. 恢复被选中进程的SP值到处理器寄存器SP
11. 恢复被选中进程的现场信息进入处理器
12. (中断返回指令触发) 逆向模式转换并弹出PSW/PC

# 进程切换的发生时机

- 进程切换一定发生在中断/异常/系统调用处理过程中，常见的情况是：
  - 阻塞式系统调用、虚拟地址异常导致被中断进程进入等待态
  - 时间片中断、I/O中断后发现更高优先级进程导致被中断进程转入就绪态
  - 终止用系统调用、不能继续执行的异常导致被中断进程进入终止态

# 进程切换与模式切换

- 一些中断/异常不会引起进程状态转换，不会引起进程切换，只是在处理完成后把控制权交回给被中断进程，处理流程是：
  - (中断/异常触发) 正向模式切换压入PSW/PC
  - 保存被中断进程的现场信息
  - 处理中断/异常
  - 恢复被中断进程的现场信息
  - (中断返回指令触发) 逆向模式转换弹出PSW/PC