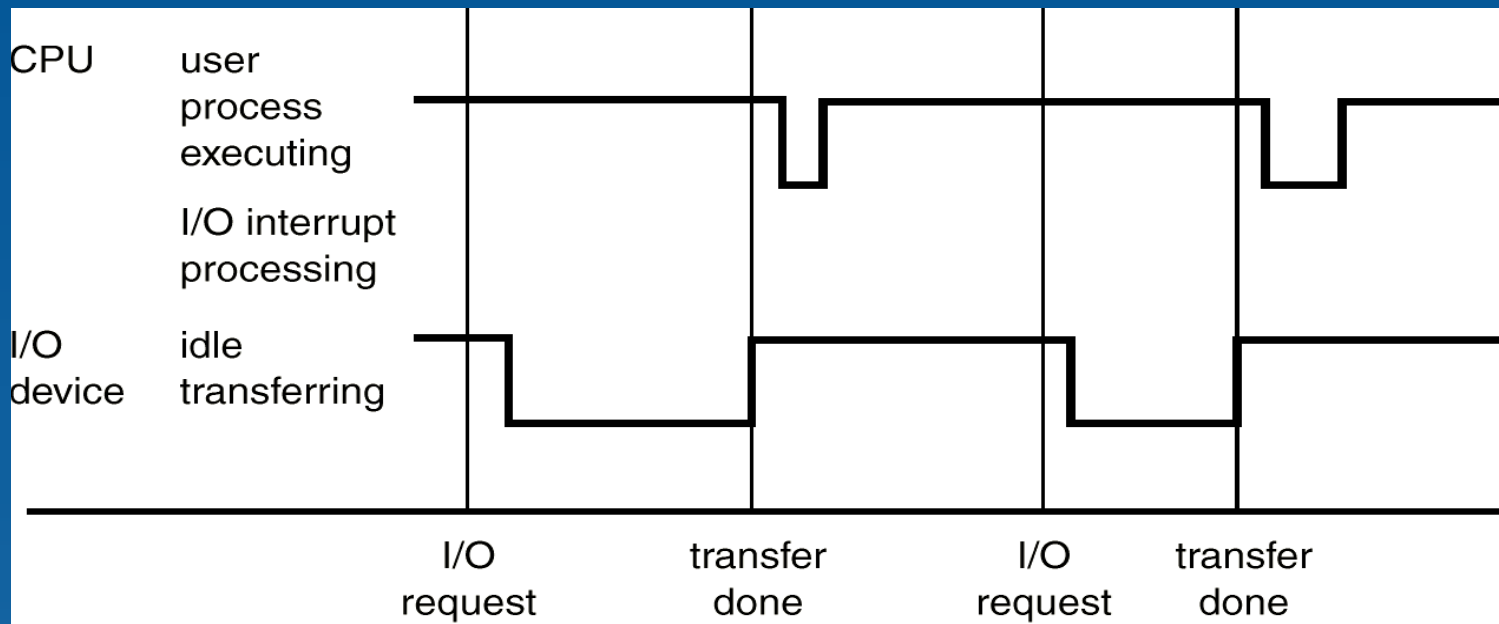




课程概念

大家熟悉的程序执行模型

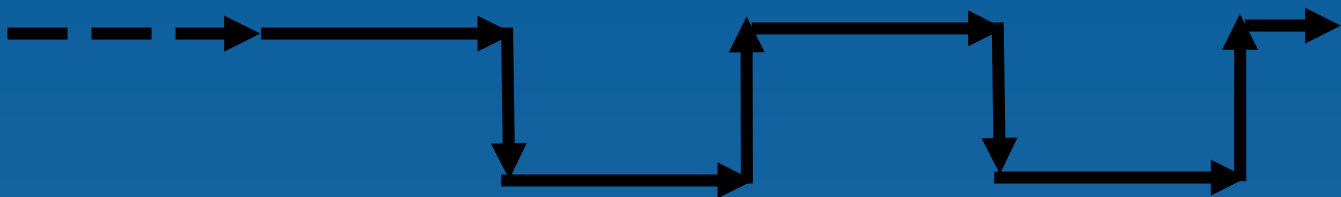


最理想的多任务模型

Program A



Program B



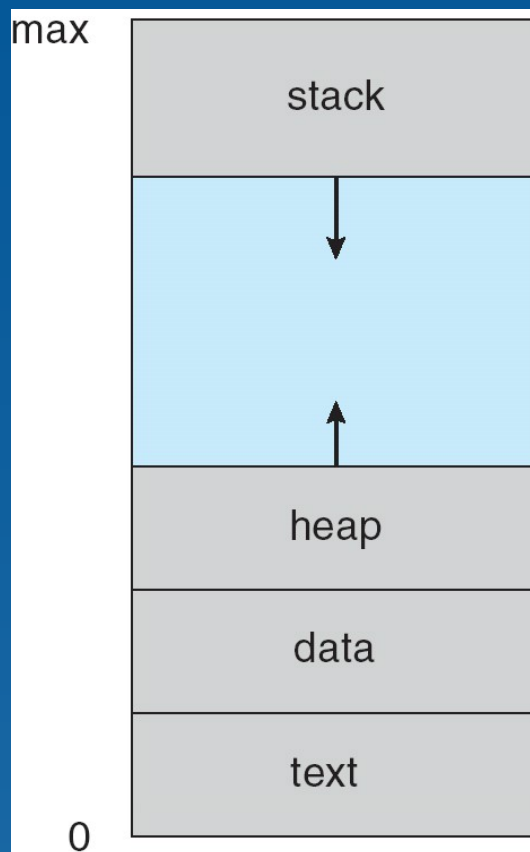
进程概念

- ◆ An operating system executes a variety of programs:
 - ◆ Batch system – jobs
 - ◆ Time-shared systems – user programs or tasks
- ◆ 为什么不能沿用“程序”？为什么“进程”？
- ◆ different data with same program
- ◆ different program with same data

进程的定义

- ◆ 教科书曾经用过 job 、 process 、 task 等词汇描述这个概念
- ◆ Process – a program in execution; process execution must progress in sequential fashion
- ◆ 进程映像 image 不仅有 program ,
- ◆ A process also includes:
 - ◆ program counter , 当前执行位置
 - ◆ Stack , 伴随的数据
 - ◆ data section , 伴随的数据

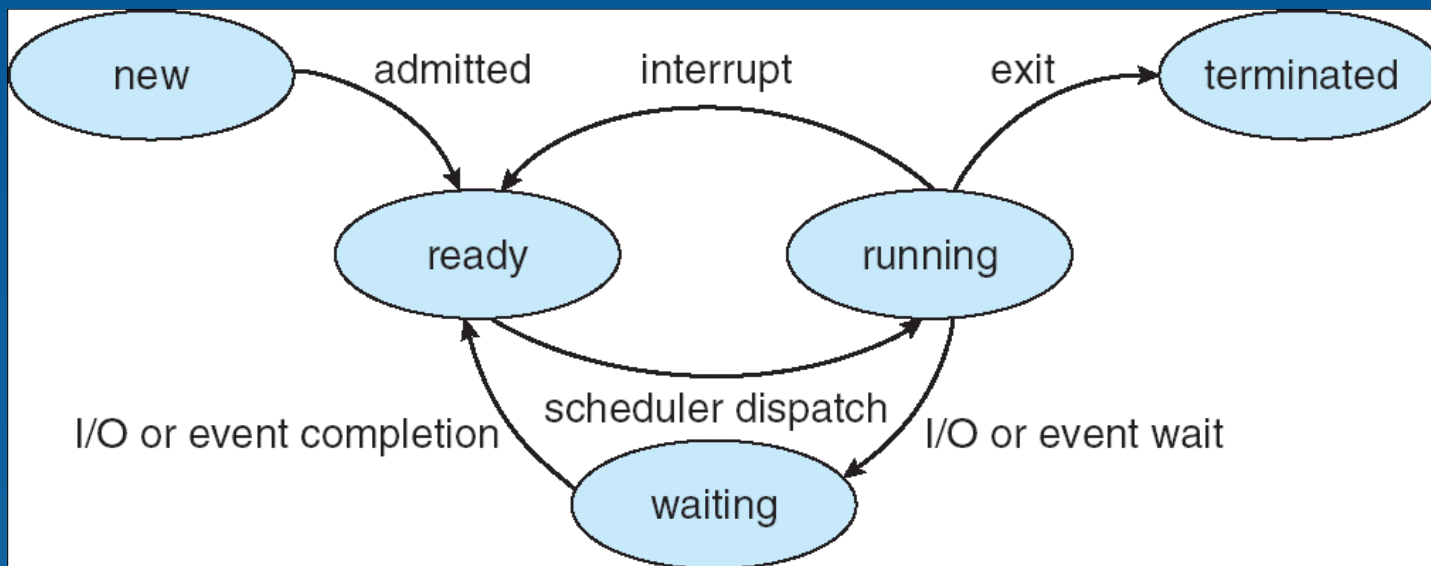
内存中的进程 Image 示例



进程状态

- ◆ 进程执行过程中，它变换着状态
 - ◆ new: 进程被创建
 - ◆ running: 进程的代码正在解释执行
 - ◆ waiting: 进程等待某个（某些）事件发生
 - ◆ ready: 进程准备就绪，
等待分配一个 CPU 来解释执行
 - ◆ terminated: 进程被终止执行

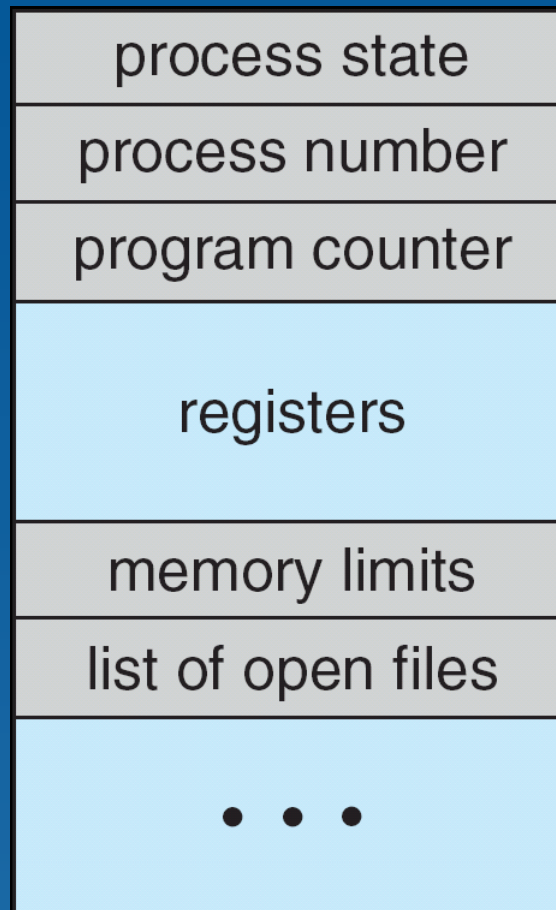
进程状态迁移示例



进程控制块 (PCB)

- ◆ 进程通常与下列信息关联
- ◆ Process state
- ◆ Program counter
- ◆ CPU registers
- ◆ CPU scheduling information
- ◆ Memory-management information
- ◆ Accounting information
- ◆ I/O status information
- ◆ THAT SPECIFIC TO THE PROCESS
- ◆ 随后以 Linux PCB 为例，“示例：Linux 的 PCB.doc”

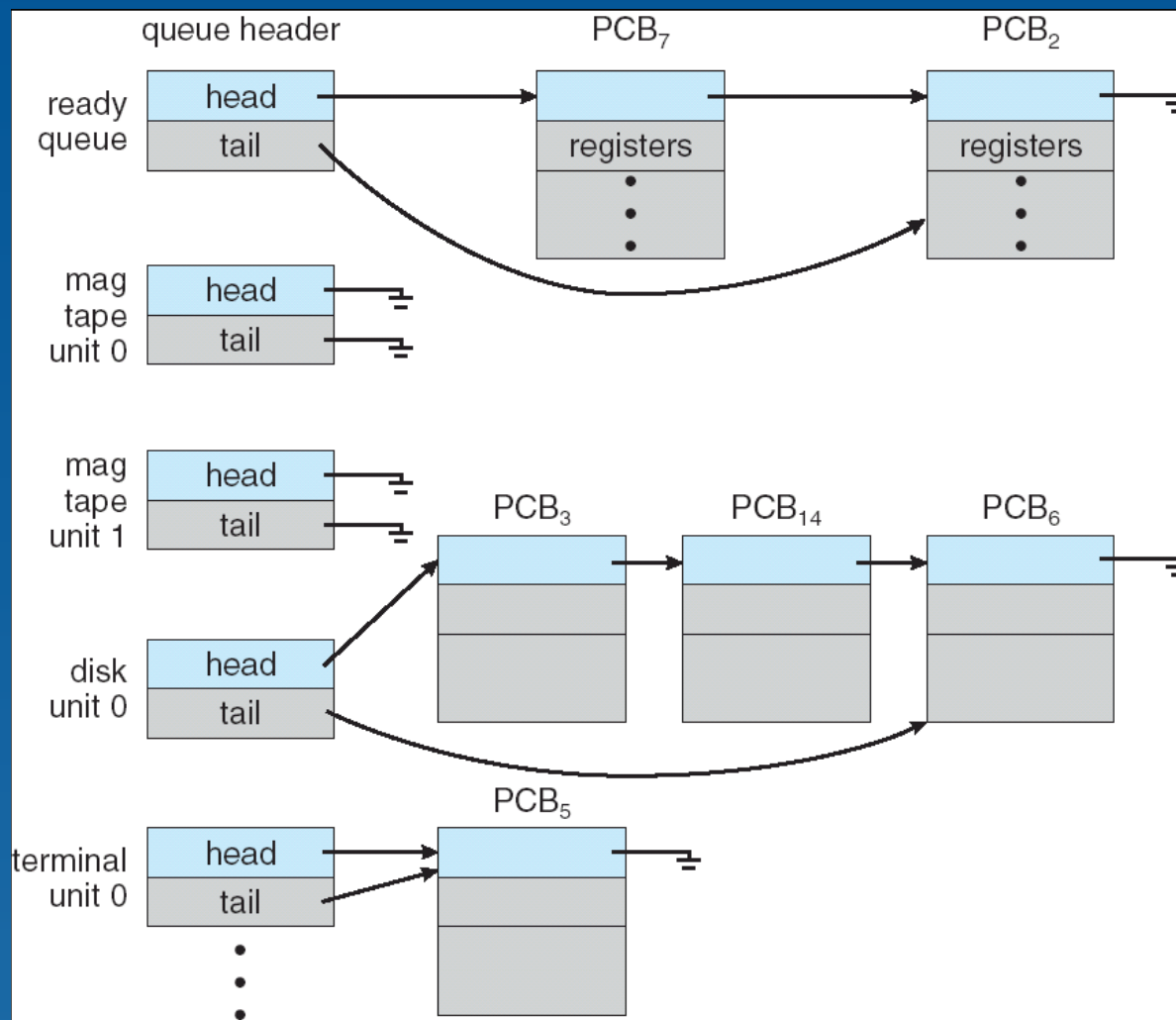
PCB 示例



进程调度队列，动态反映操作系统全貌

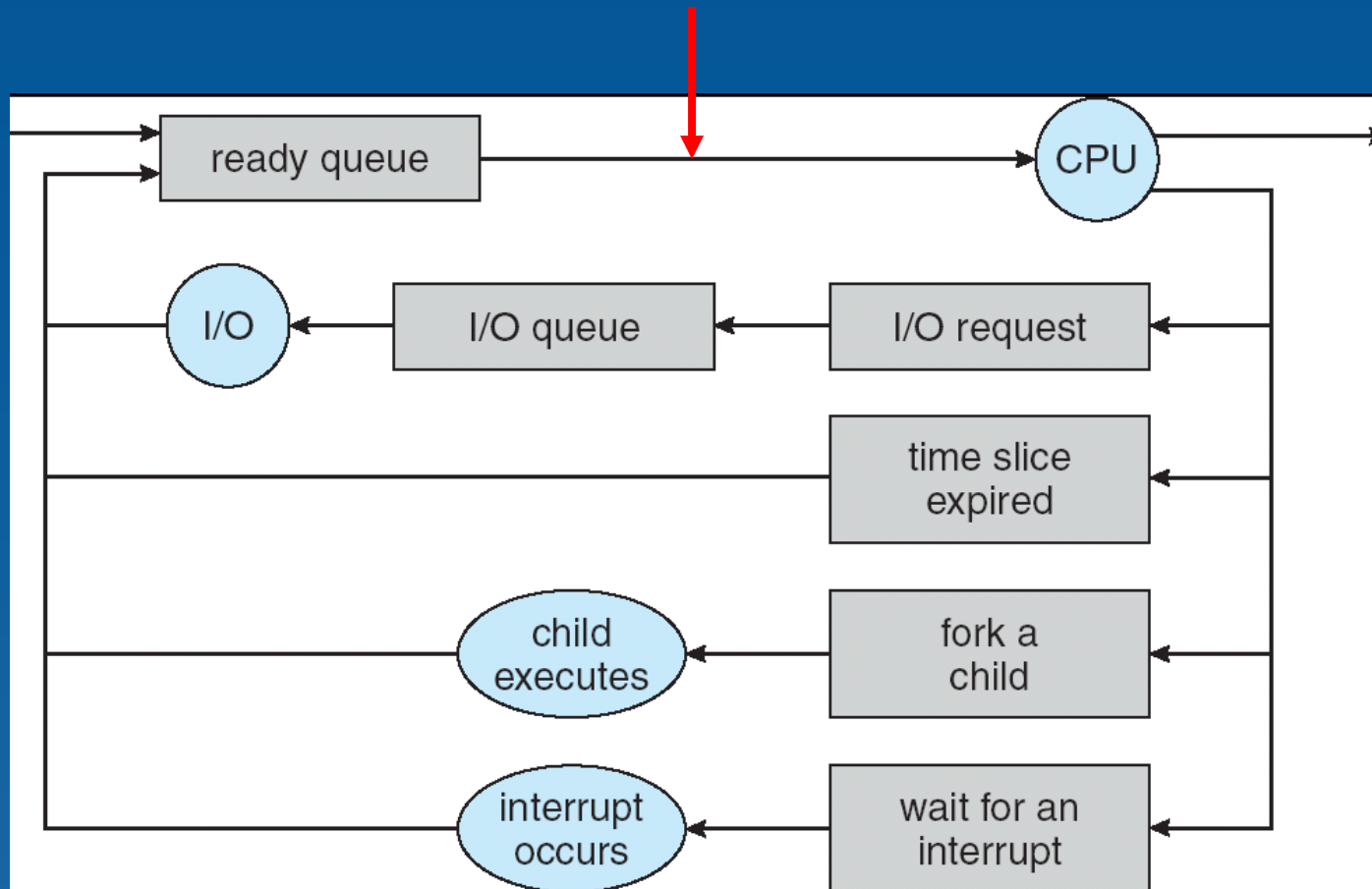
- ◆ **Job queue** – 等待进入计算机系统的待处理任务
- ◆ **Ready queue** – 驻留内存，准备就绪，等待 CPU
- ◆ **Device queues** – 等待 I/O 设备的进程
- ◆ 进程一经创建，即在这些队列之间**迁移**，直至被终止

就绪队列，各种 I/O 等待队列



一个进程在队列间迁移

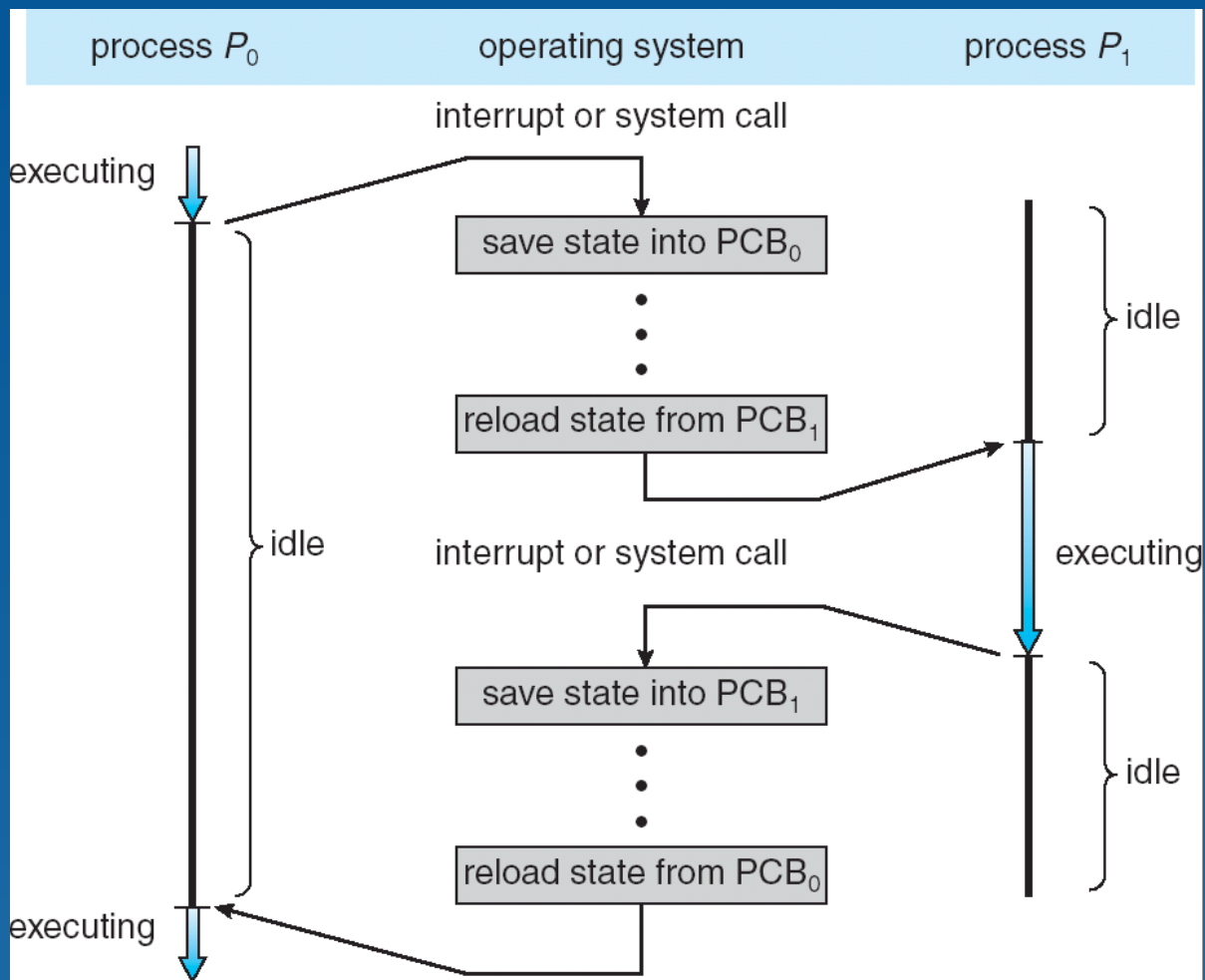
Scheduler



进程上下文切换 (Context Switch)

- ◆ CPU 任何时候只能为一个进程服务
- ◆ 当 CPU 转向为另外一个进程服务时，由于 CPU 内部资源有限，它必须保存原有（转换前）进程的状态，装入待服务（转换后）进程的状态，也即“进程上下文切换”
- ◆ “状态”指寄存器、标志位、堆栈等当前值
- ◆ 上下文切换时间是一种额外开销 (overhead)，因为期间 CPU 不做对用户进程直接有益的事
- ◆ 上下文切换时间决定于 CPU 硬件支持力度

进程切换





END