

- 用户态调用，内核态执行，注意区分，不是都在内核（比如访管指令不是特权指令）（谁都能打110)
- 用户转换到内核是硬件，不是OS
- 批处理的主要缺点是缺少交互性，分时系统才有
- 多道程序设计技术主要提高了系统利用率和吞吐量
- 时钟管理需要硬件计数器，进程调度不需要硬件
- 只能在核心态运行：广义指令（即系统调用指令），置时钟指令

## 中断过程

- 中断程序保存但子程序调用不保存：程序状态字寄存器—~~通用数据寄存器~~
- PC由中断隐指令保存，被中断程序的中断点：硬件
- OS保存通用寄存器、中断屏蔽字，初始化中断向量表

## 中断&异常区分：

- 中断（可屏蔽，不可屏蔽）：CPU执行指令之外的事：IO，时钟，DMA传送结束
- 异常（故障，自陷，终止）：CPU指令执行内部的事：访存缺页，存储保护错，除0算术溢出，陷入访管trap指令，非法操作码，地址越界

## 进程

- 不是任何时刻都有运行态进程（死锁）
- 进程的封闭性指执行结果不受外部影响：如速度不影响
- 并发不是同时执行
- 引入线程的系统中，进程不是资源调度和分派的基本单位，因为内核级线程调度分派处理器
- 进程永远是资源分配的基本单位，线程是调度的基本单位
- 父进程和子进程可并发运行，有不同PCB，不能同时使用同一临界资源，**不共享**虚拟地址空间

## 存储位置：

- 正文段：二进制代码、常量（全局变量）
- 数据堆段：动态分配存储区
- 数据栈段：临时变量，函数调用实参传递值
- PCB：进程优先级等