

### 1. 程序顺序执行的特点?

- 封闭性：程序执行的时候独占全机资源，运行环境封闭
- 可再现性：程序的运行结果可以预测和再现

### 2. 何谓进程，进程由哪些部分组成？试述进程的四大特性（动态性、独立性、并发性、结构性）及进程和程序的区别？

1. 为了描述系统中的并发活动引入的概念，进程又叫做任务，程序的一次执行过程，是程序在数据集合上顺序执行时候发生的活动

#### 2. 特性

- 动态性：程序的一次执行过程，是临时的，有生命周期的
- 独立性：系统资源分配和调度的一个独立单位
- 并发性：进程之间是并发执行的
- 结构性：PCB
  - 程序
  - 数据
  - PCB

#### 3. 区别

1. 程序是静态概念，进程是动态概念(动态性)
2. 进程是暂时的，程序是永久的(动态性)
3. 进程包含程序，数据，PCB(结构性)
4. 一个进程可以对应多个程序，一个程序可以对应多个进程，进程可以创建进程，程序不可以创建程序

### 3. 进程控制块的作用是什么?它主要包括哪几部分内容?

1. 进程的唯一，记录进程的属性信息

#### 2. 属性信息

1. 进程标识数
2. 进程状态，调度，存储器管理信息
3. 进程使用资源信息
4. CPU现场保护区
5. 记账信息
6. 进程的家族关系
7. 进程的连接指针

### 4. 进程的基本状态，试举出使进程状态发生变化的事件并描绘它的状态转换图

#### 1. 基本状态

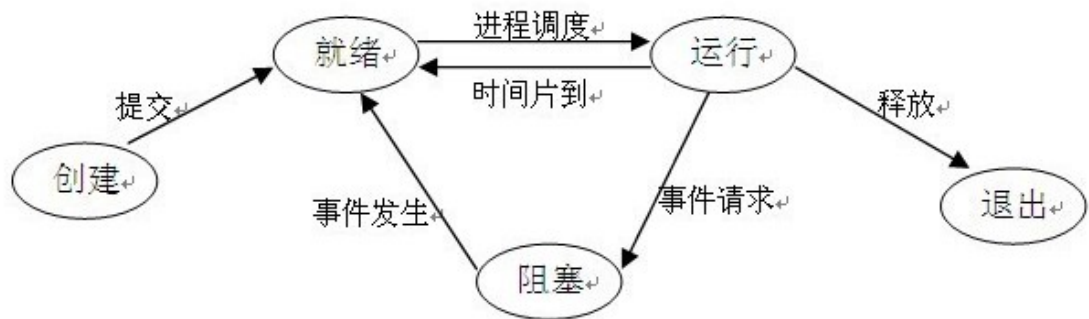
1. 创建态：进程创建初始化(大多数的操作系统都存在可以创建的进程的数目限制)
2. 终止态：进程死亡(尚未小时，等待其他进程收集相关的信息)
3. 就绪态：获得了除了CPU以外的其他全部资源，等待系统分配CPU
4. 运行态：正在CPU上执行的进程状态，一个CPU上某一时刻只能有一个进程运行
5. 阻塞态：进程等待事件的发生，即使CPU空闲也不可以运行

#### 2. 事件

1. 就绪->运行：处理机选中获得CPU
2. 运行->阻塞：程序主动的改变发生的(I/O请求)
3. 阻塞->就绪：外界事件引起的(I/O中断处理完成)
4. 运行->就绪：进程被剥夺CPU

#### 3. 转换图

## 进程三态状态装换图



### 5. 什么是原语?什么是进程控制?

#### 1. 原语

内核实现的执行过程不可中断的机器实现

#### 2. 进程控制

是指系统使用一些具有特定功能的程序段(原语)来创建、撤消进程, 以及完成进程各状态之间的转换

### 6. 进程调度的功能、方式、时机、算法。作业调度, 交换调度。作业的周转时间和作业的带权周转时间?

#### 1. 功能

- 管理系统中进程的执行状况, 状况信息保存在CPU中
- 选择进程的真正的占有CPU

\* 进程的上下文切换

#### 2. 方式

- 抢险式: 批处理系统
- 非抢先式: 分时和实时系统

#### 3. 时机

- 自身
  - 进程完成或者进程终止
  - I/O请求
  - 进程执行原语操作
- 系统
  - 时间片轮转
  - 优先级调度

#### 4. 算法

- 先来先服务
- 最短优先
- 响应比高者优先
- 优先级调度
- 轮转法
- 多级反馈队列轮转法
- 实时系统
  - 时钟驱动
  - 加权轮转

#### 5. 作业调度

## 6. 交换调度：

将处于主存就绪或者主存阻塞暂时不具备运行条件的进程换出道外存交换区，并将外存中具有运行条件的进程换入主存

7. 作业周转时间：作业完成时间-作业提交时间

8. 作业带权周转时间：周转时间/服务时间

9. 平均周转时间=作业周转总时间/作业个数

10. 平均带权周转时间=带权周转总时间/作业个数

11. 线程的定义，线程与进程的比较。系统对线程的支持（用户级线程、核心级线程、两级组合）。

### 1. 线程定义

- 线程是进程的子任务
- 线程是进程中的一个可执行实体，操作系统调度的独立单位
- 线程组共享进程的资源

### 2. 比较

- 资源：进程拥有独立的存储空间，线程共享进程资源但是也独立拥有一部分所有资源
- 调度：进程上下文切换代价高，线程代价效率高
- 并发性：线程可以提高并发性
- 安全性：线程危害安全性，但是共享数据方便

### 3. 线程支持

1. 用户级线程
2. 核心级线程
3. 两级组合

12. 并发执行的进程在系统中通常表现为几种关系？各是在什么情况下发生的？

1. 互斥：共享资源，间接的制约关系
2. 同步：互相协作的直接制约关系
3. 前序：互斥和同步关系决定了进程的前序关系

13. 什么叫临界资源？什么叫临界区？对临界区的使用应符合的四个准则（互斥使用、让权等待、有空让进、有限等待）。

1. 临界资源：一次仅允许一个进程使用的系统中共享资源
2. 临界区：并发进程访问临界资源的那段必须互斥执行的程序
3. 四个准则

- 互斥使用：不存在两个程序在临界区执行
- 让权等待：进入临界区放弃处理机阻塞
- 有空让进：不可阻止进程进入临界区
- 有限等待：有限等待时间

14. 解决进程之间互斥的办法：开、关中断，加锁、开锁（又叫测试与设置，通常由一条机器指令完成），软件方法，信号量与P、V操作

1. 中断：进程在临界区执行的时候，关闭所有的中断
  - 实现简单
  - 代价过高，限制了程序的交叉处理能力

### 2. 加锁开锁

测试和设置指令是一条不可中断的机器指令

- 实现简单有效
- 进程循环测试锁位，CPU时间被浪费，忙等待

### 3. 软件实现方法

### 4. 信号量

15. 若信号量S表示某一类资源，则对S执行P、V操作的直观含意是什么？当进程对信号量S执行P、V操作时，S的值发生变化，当 $S > 0$ 、 $S = 0$ 、和 $S < 0$ 时，其物理意义是什么？

1. P

- 申请资源，等待
- `value--`

2. V

- 释放资源，发送信号
- `value++`

3. S

- $S > 0$  : 贡献资源存在空闲
- $S = 0$  : 贡献资源0
- $S < 0$  : 绝对值表示正在阻塞等待的进程数

16. 在用P/V操作实现进程通信时，应根据什么原则对信号量赋初值？

????????????????????????????????

17. 经典的IPC问题。

18. 进程高级通信有哪些实现机制？

1. 消息缓冲
2. 信箱
3. 管道
4. 共享内存

19. 死锁产生的必要条件及解决死锁的方法

1. 必要条件

1. 互斥条件
2. 保持等待条件
3. 不剥夺条件
4. 循环等待条件

2. 方法

1. 鸵鸟算法
2. 死锁预防
  1. 破坏互斥条件
  2. 破坏保持等待条件
  3. 破坏不剥夺条件
  4. 破坏循环等待条件
3. 死锁避免
4. 死锁检测和恢复

20. 理解银行家算法的实质。能够利用银行家算法避免死锁