

日期:

贾星宇 202000300125

## 一. 操作系统

1. 计算机系统自下而上大致分为：硬件、操作系统、应用程序、用户
2. 操作系统：控制和管理工作与资源的分配，合理地组织、调度计算机的工作与资源的分配，进而为用户和其它软件提供方便接口与环境的程序集合。

是计算机系统中最基本的系统软件。

### 3. 操作系统的特征，

并发、共享、虚拟、异步

Concurrence, Sharing, Virtual, Asynchronism.

- (1) 并发：两个或多个事件在 同一时间间隔内 发生。

并行、同一时刻

- (2) 共享：系统中资源可供内存中多个并发执行的进程共同使用。

{ 互斥共享方式：一段时间内只允许一个进程访问

{ 同时访问方式：宏观同时，微观上多个进程交替访问

- (3) 虚拟：一个物理上的实体 → 若干逻辑上的对应物

{ 虚拟处理器：多道程序并发执行，分时使用一个处理器  
(时分复用技术)

{ 虚拟存储器：物理存储 → 虚拟存储，逻辑上扩充容量。

(空分复用技术)

- (4) 异步：进程的执行受资源影响，走走停停

## 4. 操作系统功能：处理机管理、存储器管理、设备管理、文件管理

向用户提供接口

扩充机器

- (1) 计算机系统资源的管理者，

处理机

存储器

文件

设备



日期: /

(2) 作为用户与计算机硬件系统之间的接口。

① 命令接口 { 联机命令接口: 交互式命令接口, 输入-系统执行-输出  
脱机命令接口: <sup>分时/实时系统</sup>提供一组作业控制命令  
<sub>批处理系统</sub>

② 程序接口: 由一组系统调用 (广义指令) 组成

(3) 实现了对计算机资源的扩充

5. CPU 执行的程序:

{ 操作系统内核程序 (管理程序) 在管态运行  
用户自编程序 (应用程序) 在目态运行

指令: { 特权指令: 不允许用户直接使用 (I/O, 置中断 ...)  
非特权指令: 不能直接访问系统中的软硬件资源, 仅限于访问用户的地址空间

CPU 运行模式 { 用户态 (目态)  
核心态 (管态、内核态)

△ 切换到用户态的指令也是特权指令

6. 操作系统内核包括:

(1) 时钟管理: 计时、进程切换 (时钟中断) ...

(2) 中断机制: 提高多道程序运行环境中 CPU 的利用率

(3) 原语: 关闭中断, 让其所有动作不可分割地完成后再打开中断

(4) 系统控制的数据结构及处理

7. 原语: 1) 处于操作系统最低层, 最接近硬件的部分

2) 程序运行具有原子性, 操作只能一气呵成

3) 程序运行时间较短, 且调用频繁



日期: /

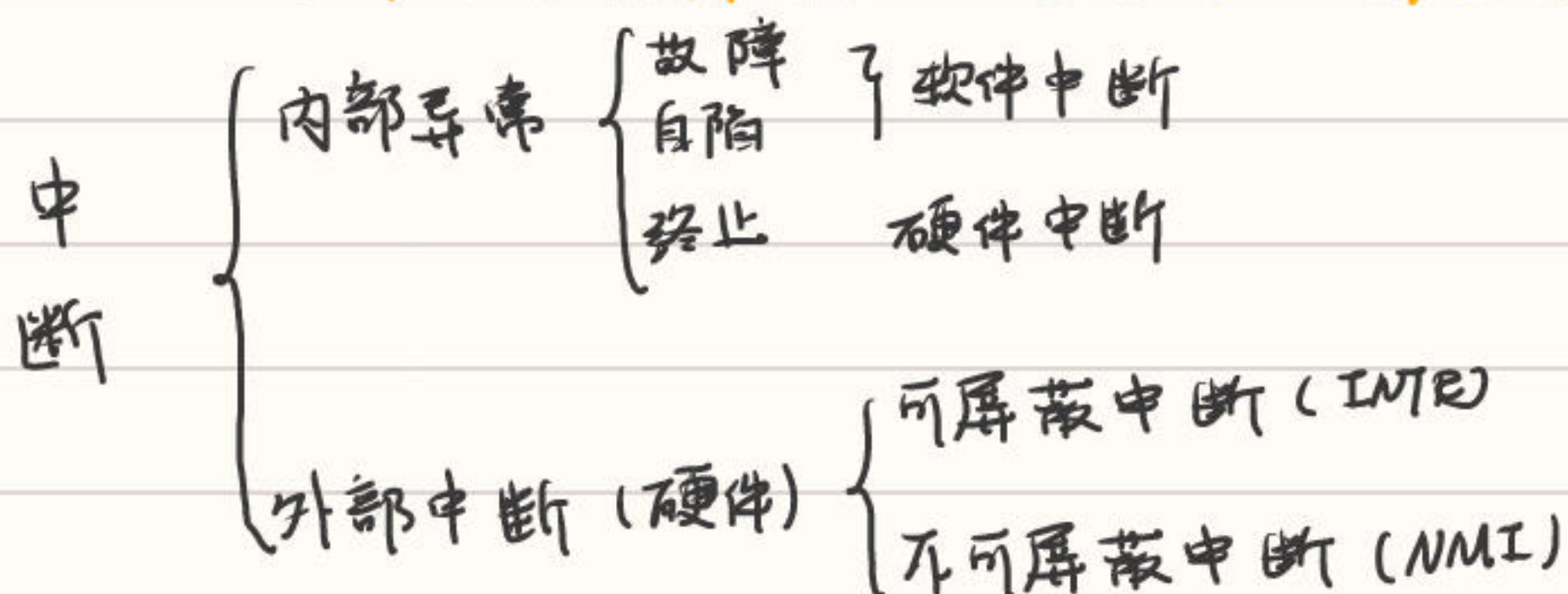
8. 中断, 也称外中断, 来自CPU执行指令外部的事件

设备发出的 I/O 结束中断, 时钟中断 (一个时间片已到) ...

异常, 也称内中断, 来自CPU执行指令内部的事件

程序非法操作码, 地址越界, 运算溢出 ...

异常不能被屏蔽, 一旦出现应立即处理.



用户态 → 核心态 通过中断实现, 且中断是唯一途径

核心态 → 用户态 通过执行某特权指令.

9. 系统调用: 用户在程序中调用操作系统所提供的一些子功能

- 是特殊的公共子程序.

- 凡是与资源有关的操作 (存储分配, I/O 传输, 管理文件) 都必须通过系统调用方式向操作系统提出服务请求, 并由操作系统代为完成.

- 系统调用必定需要使用某些特权指令完成.

10. 用户可以执行 陷入指令 (访管指令, trap 指令) 发起系统调用

—— 相当于把 CPU 使用权主动交给操作系统

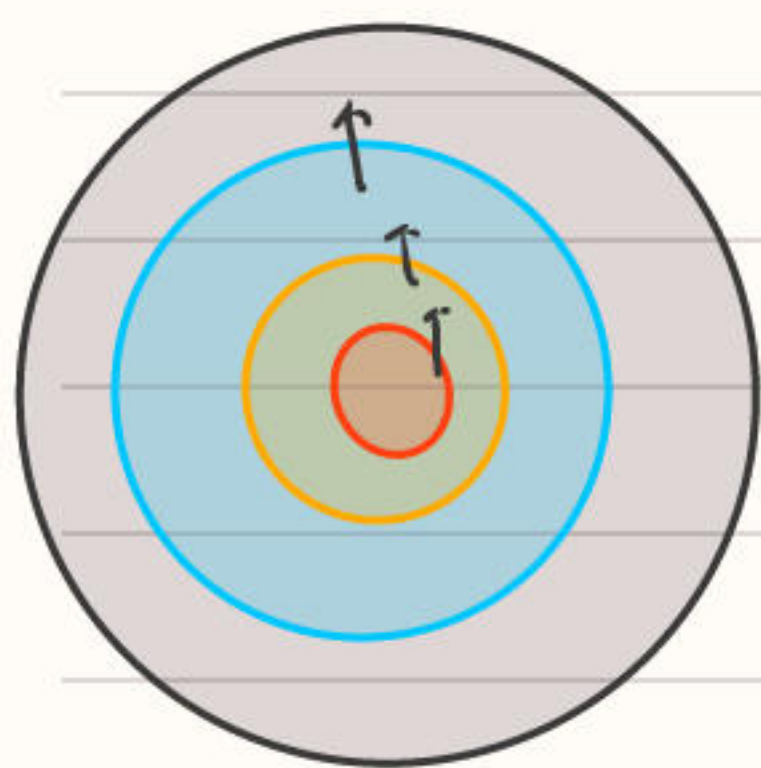
11. 操作系统结构

11.1 分层法: 最底层为硬件, 最高层为用户接口

每层仅能调用紧靠它的低层的功能和服务.

优: 便于系统调试验证, 易扩充, 易维护

缺: 难以合理定义, 效率差.





日期: /

(2) 模块化. 将系统划分为若干具有一定独立性的模块

规定好各模块间的接口

每个模块也可划分为若干具有一定功能的子模块

同样规定好各子模块间的接口.

○做到模块内高内聚, 模块间低耦合.

优: 提高了正确性、可理解性、可维护性、可适应性

加速了开发过程

缺: 接口难以满足实际需求

各模块同时设计, 每块决定无法建立在上一块基础上.

(3) 宏内核. (从操作系统内核架构划分)

单内核/大内核. 将系统主功能模块作为一个紧密联系的整体运行在核心态.

(4) 微内核. 将内核中最基本的功能保留在内核, 其余移到用户态执行

将操作系统划分为: ①微内核: 与硬件处理紧密相关的部分, 较基本的功能、客户和服务端间的通信

②多个服务器: 进程/线程管理、虚拟存储器管理.

客户与服务端之间借助微内核提供的消息传递机制实现交互

(5) 微内核的基本功能

①进程/线程管理

②低级存储器管理 (逻辑地址 → 物理地址)

③中断和陷入处理 (负责识别, 随后交给相关服务器处理)

(6) 微内核特点:

扩展性、灵活性

可靠性、安全性

可移植性

分布式计算

性能问题: 核心态与用户态间切换, 开销较大.



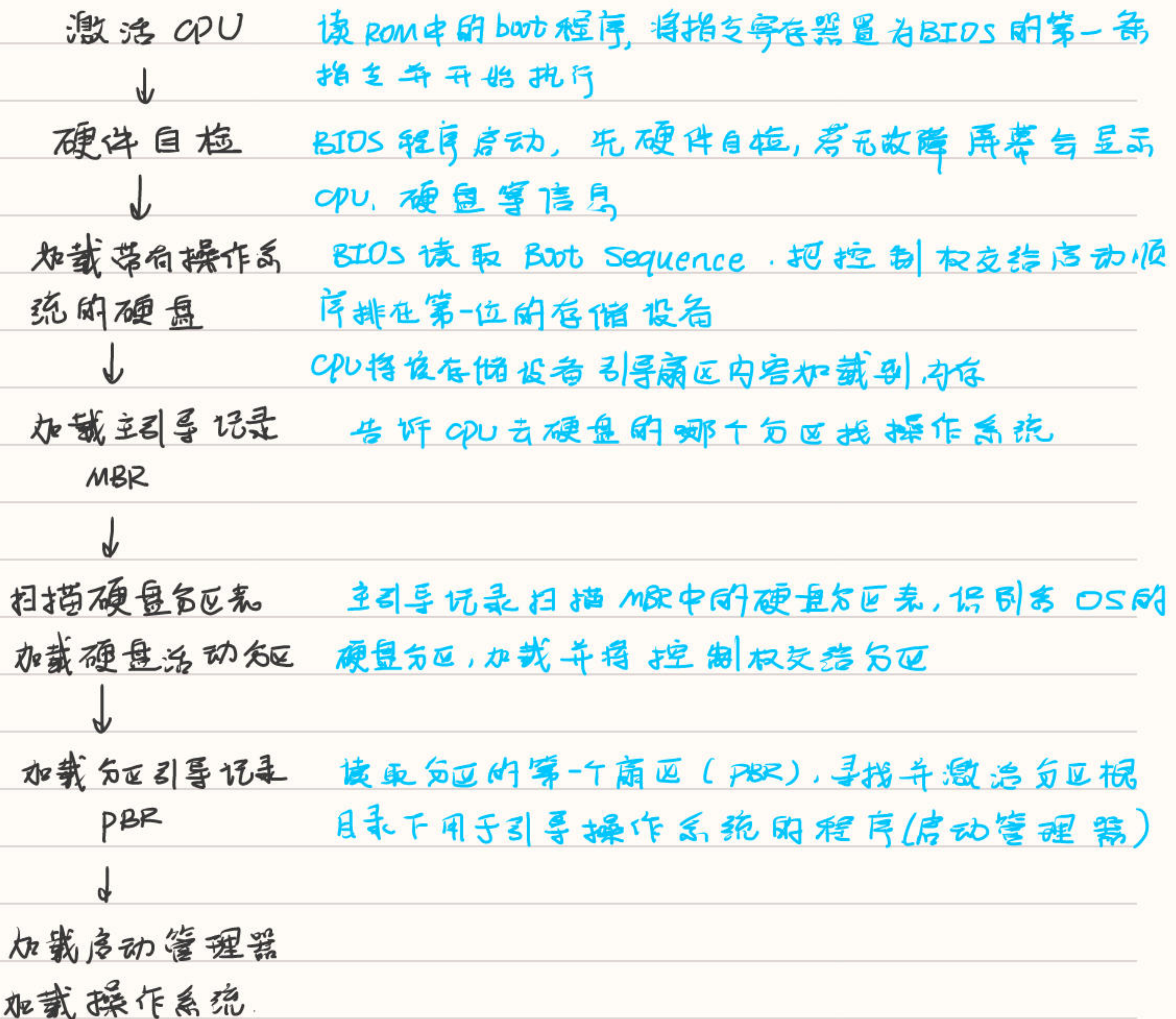
日期: /

(7) 对机器进行分区——外核。

每个用户拥有整个资源的子集

在底层 外核(一个程序) 在内核会运行, 为虚拟机分配资源, 检查使用这些资源的正图。

## 12. 操作系统引导:



## 13. 虚拟机:

是一台逻辑计算机。指利用特殊的虚拟化技术, 通过隐藏特定计算平台的实际物理特性, 为用户提供抽象的、统一的、模拟的计算环境。



日期: /

## 14. 虚拟管理程序分类

(1) 第一类: 唯一运行在最高特权级的程序

在裸机上运行, 并且具备多道程序功能.

① 虚拟机作为用户态的一个进程运行, 不允许执行敏感指令

② 其上的操作系统认为自己处于内核态 —— 虚拟内核态

(2) 第二类: 依赖于操作系统分配和调度资源的程序.

## 二. 进程

1. 为了更好地描述和控制程序的并发执行, 实现操作系统的并发性和共享性.

2. 进程控制块 (PCB, Process Control Block):

描述进程的基本情况和运行状态, 进而控制和管理进程.

进程实体 (进程映像):

程序段、相关数据断、PCB

进程映像 **是静态的**, 进程 **是动态的**

3. 撤销进程, 实质上是撤销进程的 PCB

PCB 是进程存在的唯一标志

## 4. 进程的特征:

动态性: 进程是程序的一次执行, 有创建、活动、暂停、终止等过程. 是进程最基本的特征.

并发性: 多个进程实体同存于内存中, 能在一段时间内同时运行.

独立性: 进程实体是一个能独立运行、独立获得资源和独立接受调度的基本单位

异步性: 各进程按各自独立的、不可预知的速度向前推进.

↙ 它会导致执行结果的不可再现性



日期: /

## 5. 进程的状态与转换,

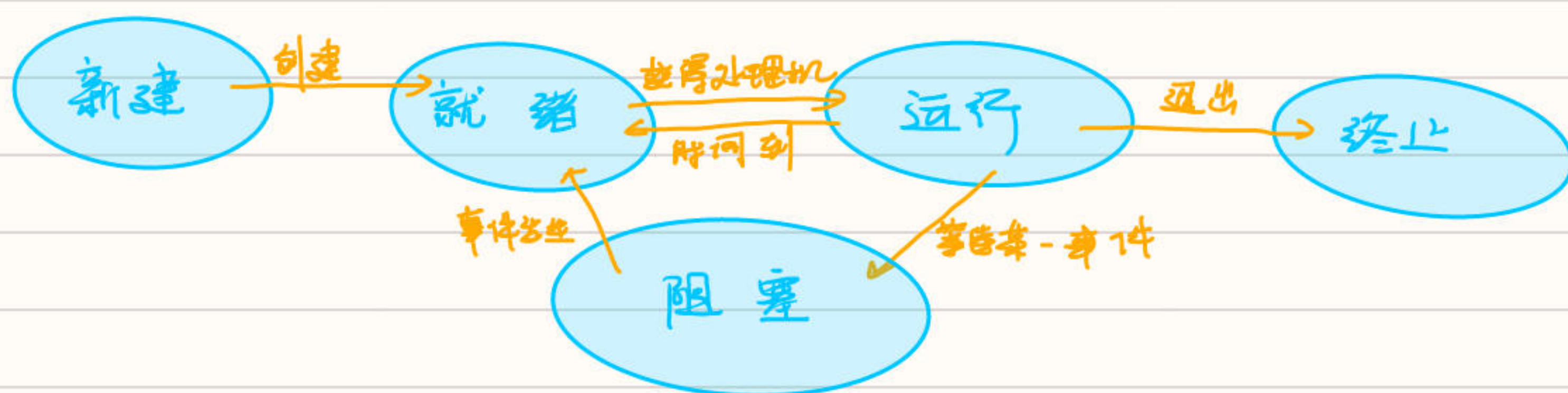
运行态: 每时刻每个处理机上只有一个进程处于运行态

就绪态: 进程获得了除处理机外的一切所需资源

阻塞态 (等待态): 等待某资源为可用 (处理机外) 或等待输入/输出完成等.

创建态: 正在创建, 尚未成就就绪态

结束态: 正从系统中消失



## 6. 进程控制块 (PCB): 进程最核心的部分.

(1) PCB 是进程 (实体) 的一部分, 是进程存在的唯一标志.

(2) 进程创建 — 新建 PCB

进程执行 — 系统通过其 PCB 了解进程的现行状态信息

进程结束 — 系统收回 PCB.

(3) 主要包括:

① 进程描述信息 (进程/用户标识符)

② 进程控制和管理信息 (当前状态、优先级……)

③ 资源分配清单 (有关内存/虚拟地址空间的情况)

④ 处理机相关信息 (处理机的上下文, 即各寄存器的值等)

(4) 各进程 PCB 组织方式:

① 链接方式, 将同一状态 PCB 链接成一个队列

② 索引方式, 将同一状态的进程组织在一个索引表中.



日期: /

7. 程序段: 能被进程调度程序调度到 CPU 执行的程序代码段。

程序可被多个进程共享

8. 数据段: 进程对应程序加工处理的原始数据, 程序执行时产生的中间或最终结果。

9. 进程的通信:

(1) 共享存储

① 通信的进程之间存在一块可直接访问的共享空间

② 使用同步互斥工具: 同一时刻仅可读/写

③ 低级方式: 基于数据结构的共享

高级方式: 基于存储区的共享。

两个进程共享空间, 必须通过特殊的系统调用实现。

(2) 消息传递:

① 进程间数据交换以格式化消息为单位

通过系统提供的发送消息和接收消息两个原语进行数据交换

② 分类: 直接通信方式: 发送进程直接把消息发送给接收进程  
挂在接收进程的消息缓冲队列中

间接通信方式: 发送进程把消息发给某中间实体 (信箱)

(3) 管道通信。

① 管道: 连接一个读进程和一个写进程以实现他们之间通信的一个共享文件 (pipe 文件)

② 未写满读功能阻塞, 未读空写功能阻塞

③ 读数据是一次性操作, 一旦读取便释放空间

④ 半双工通信: 某-时刻只能单向传输。



日期: /