

# 第一章 操作系统引论

## 1. 操作系统 是方便用户、管理和控制计算机资源软件

Operating System (OS)

## 2. 操作系统的目标

方便性：使计算机易于使用（用户的角度）

有效性：使计算机资源利用率、吞吐率高（系统管理员）

可扩充性：可以方便地进行更新修改以运新硬件，易于开发（开发人员角度）

开放性：可以实现互连

是用户与计算机硬件系统之间的接口。

是计算机资源的管理者。

实现了对计算机资源的抽象。

## 3. 操作系统的作用

## 4. 操作系统的发展

人工操作方式

用户独占资源

CPU等待人工操作（慢）

CPU利用率低（慢）

子道

子道

子道

解译输入

解译输出

输入

解译输入输出

（批处理技术）

脱机输入输出方式

加入了外围机代替人工操作

减少了CPU空闲时间

提高了资源利用率，吞吐量增大

单道批处理系统

由于内存中只允许存放一道作业，资源利用率低

一道作业，资源利用率低

资源利用率低

革命性变革

多道批处理系统

多道作业能同时运行，提高了利用率

但对资源利用率仍不高

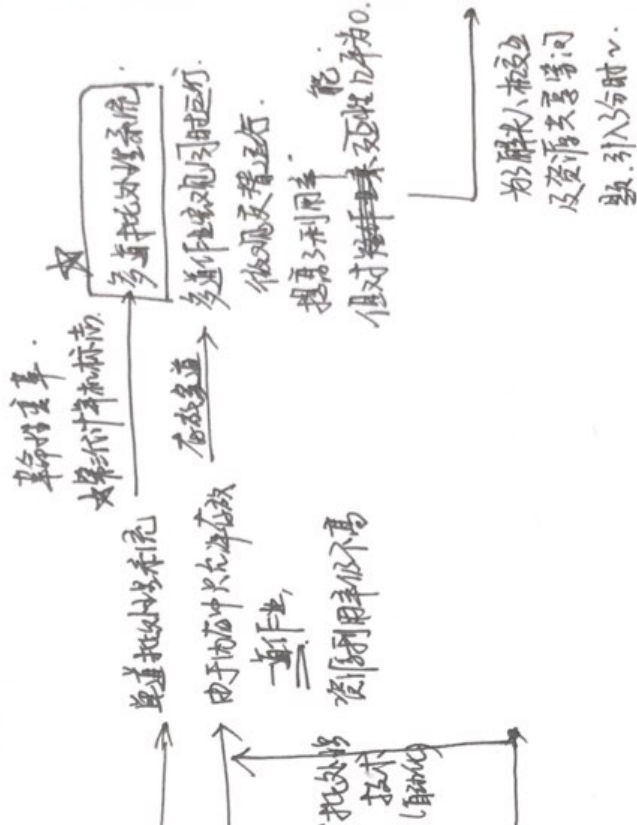
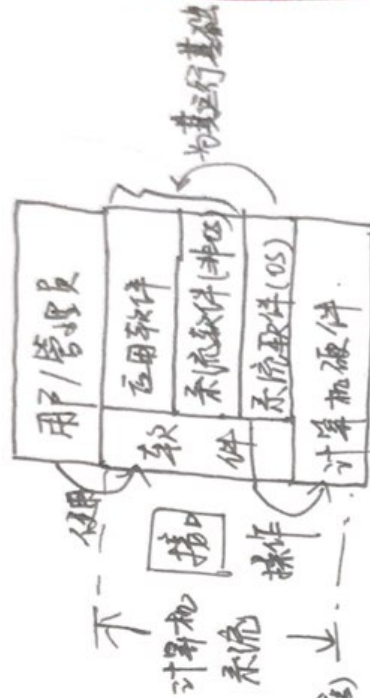
资源利用率低

多道批处理系统

多道作业能同时运行，提高了利用率

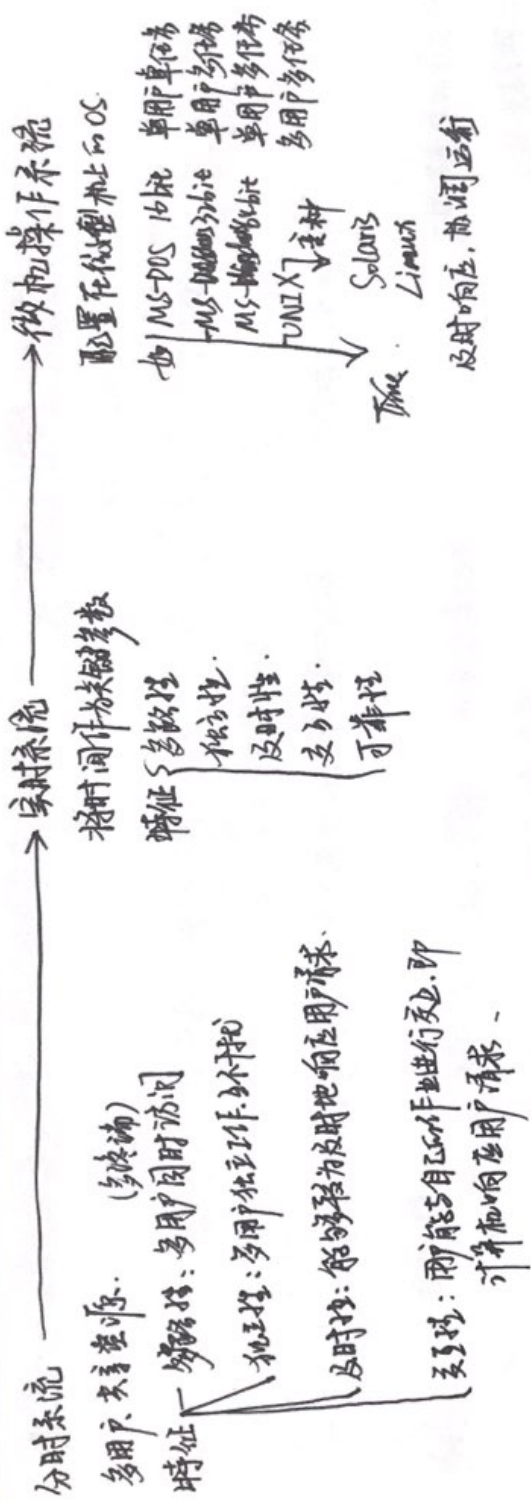
但对资源利用率仍不高

资源利用率低



批处理技术是指在系统中配置一个监督程序，由其控制一批作业自动处理的技术。

为解决人机交互及资源共享等问题，引入了分时系统。



Q: 分时系统和实时系统的区别?

- A:
- ① 交互性: 分时系统交互性广泛, 实时系统交互性仅限于访问系统中某些特定应用程序。
  - ② 及时性: 分时系统 实时 实时对及时性要求较高
  - ③ 可靠性: 分时 < 实时, 实时 ~ 可靠性较好。

## 5. 操作系统的特征:

### 并发性 (宏观):

指两个或多个事件在同一时间间隔内发生。

### 共享性:

~ 点 ← → 并行性

系统中的资源可供多个并发执行的进程同时使用。

### 虚拟性:

物理实体  $\xrightarrow{\text{虚拟技术}}$  集对应用 (逻辑上), 即资源虚拟化。

目的: 使进程可以更方便地共享系统资源。

### 异步性:

前提: 多道程序

未知: 如何运行 (何时执行, 何时暂停), 如上图中, 执行时间可能不同。

已知: 执行结果相同。



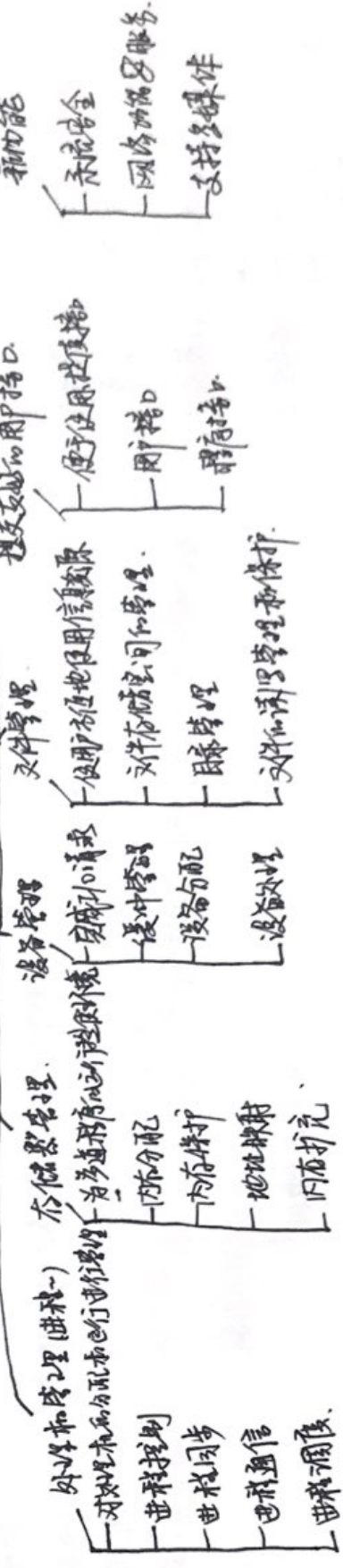
### 不同操作系统的结构图

- ① 无结构 OS
- ② 模块化结构 OS
- ③ 分层式结构 OS
- ④ 微内核结构 OS

变革

## 6. 操作系统的功能:

### 功能





## 第2章 进程的描述与控制

### 1. 进程的基本概念



意味着不存在循环

#### ① 程序:

(i) 控制流图: 用来表示程序的执行顺序, 是有向无环图(DAG), 可以用  $P_1 \rightarrow P_2$  来表示, DAG图如上.

#### (ii) 执行方式:

概念: 若干程序(或)必须严格按照先后次序执行, 如执行完  $P_1$ , 才能执行  $P_2$ .

顺序执行 { 概念: 顺序性: 按照顺序执行  
特征: 封闭性: 执行时独占资源  
可再现性: 只要初始条件相同, 重复执行结果相同

概念: 两个或以上程序(或)可在同一时间同时执行(同时).

并发执行 { 概念: 提高了资源利用率和吞吐量  
优点: 提高了资源利用率和吞吐量  
问题性: 由于资源竞争和相合作, 并发执行的程序间形成相互制约的关系, 出现“执行-等待-执行”  
阻塞性: 并发执行的程序本身所执行的指令不受外界其他程序的影响  
不可再现性: 初始条件相同, 重复执行结果不一定相同

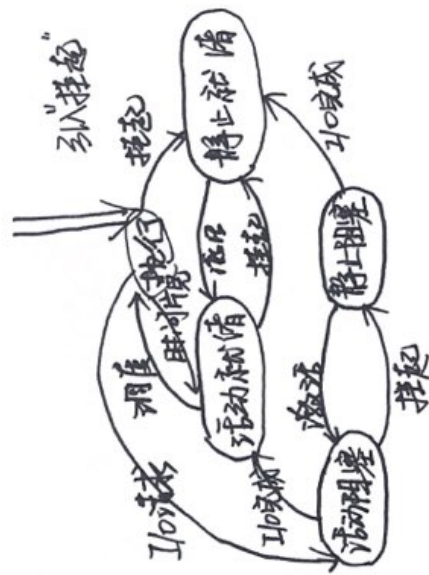
#### ② 进程

(i) 引入原因: 程序并发执行的不可再现性: (并行性)  
(ii) 进程与程序的区别: 进程是程序的一次运行, 程序是静态的文件, 而进程是动态的运行.  
(iii) 进程实体(简称进程): PCP (进程控制块)  
包含进程标识符、地址、状态、进程间通信信息、进程控制信息

程序段

数据段  
堆  
栈  
共享数据区

特征 { 连续性: 进程程序的执行.  
并发性: 多个进程在系统中同时执行.  
独立性: 可独立运行, 独立分配资源.  
异步性: 进程可接收到的, 不可预期的事件.  
后发性: 但可再发性, 再保证



控制:

主要目标: 创建和终止进程以及负责进程迁移过程中状态转换等功能。

主要功能: 创建和终止进程、在系统内迁移、原址性。

实现方式: 原语 (该语言不允许中断, 常称为阻塞语言)

执行一切命令

操作系统：高姿态 执行一切命令  
操作系统：低姿态 执行期望的命令

执行规定的命令

### 3. 进程同步

指对多个相逐进程在执行次序上进行协调, 使程序的执行具有可再现性.

① 制约关系: 资源竞争  $\rightarrow$  互斥相制约 (竞争等待资源)

进程合作  $\rightarrow$  直接相制约 (前驱阻塞)

② 临界资源和临界区.

ii) 临界资源: 每次只允许一个进程使用.

iii) 临界区: 访问临界资源的代码片段.

$\Rightarrow$  因此, 在互斥地使用临界资源.

③ 同步机制的规则 (实现互斥)

i) 空闲让进 ii) 忙则等待

iii) 有限等待  
避免“死等”

iv) 让权等待  
避免“忙等”  
对cpu

④ 信号量机制 (同步机制)

ii) 发展.

整型信号量

记录型信号量.  $\rightarrow$  信号量集.

引入阻塞队列

避免“让权等待”

$P(wait(s)) \quad V(signal(s))$

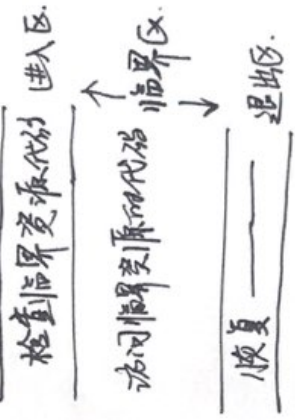
存在问题: 未遵循“让权等待”

ii) 应用: 实现前驱关系; 实现互斥

iii) 经典问题: 生产者-消费者 ~

哲学家就餐 ~

读者-写者 ~





#### 4. 进程通信.

进程间的信息交换.

- ① 类型 {
- 共享存储系统
  - 管道通信
  - 消息传递系统
  - 客户机-服务器系统

- ② 本例 {
- 数据结构
  - 发送原语
  - 接收原语.

#### 5. 线程.

引入目的: 减少程序在并发执行时所付出的时空开销.

线程控制与进程控制类似

线程模型 {

- 用户级线程
- 内核级线程
- 混合方式.

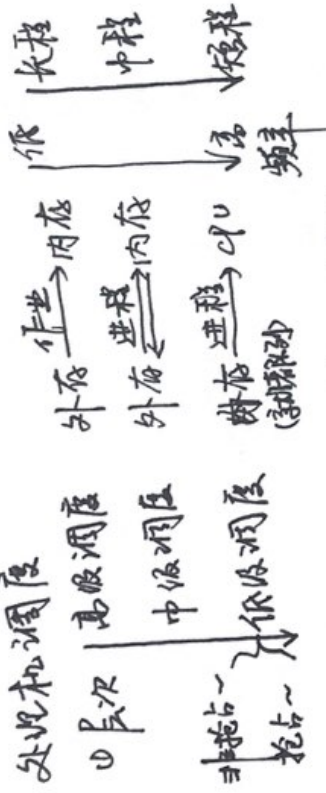
Q. 线程和进程的区别?

- A. ① 一个进程含有一个或多个相对独立之的线程.  
② 进程只是拥有资源的基单位, 线程是可执行实体.  
③ 线程是CPU调度和分派的基本单位.

## 第3章 处理器调度与死锁:

往往有进程数 > 处理器数目 ∴ 引入处理器调度

### 1. 处理器调度



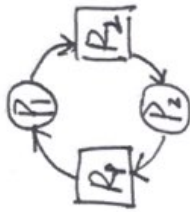
② 目标: 吞吐量、利用率、公平性

③ 调度算法

- 先来先服务算法 (FCFS)
- 短作业优先 ~ (SJF)
- 优先级调度 ~ (PSA)
- 高响应比优先调度算法 (HRRN)
- 时间片轮转法 (RR)
- 多级队列调度算法
- 多级反馈队列调度算法 (FB)
- 最早截止时间优先算法 (EDF)
- 最低松弛度优先算法 (LLF)

实时~





## 2. 死锁:

① 定义: 某组进程 (进程集合) 中的每个进程都在等待另一个进程释放资源 (处于等待状态)

② 原因: (i) 资源竞争 (ii) 进程推进顺序非法

③ 产生 ~ 的必要条件:

冲突条件 { 互斥条件: 资源互斥  
请求与保持条件: 资源请求, 保持.  
不剥夺条件 (不可抢占)  
环路等待条件: 构成环路, 如右图.

④ 处理死锁方法:

方法 { 预防死锁 — 破坏相应条件 { 安全状态 (对某一时刻而言)  
避免死锁 — 资源动态分配策略 { 银行家算法 (安全性算法)  
检测死锁  
解除死锁

第四章 存储管理

1. 存储数据层结构:

10. 自反同: 希望容量无限大. 速率无限快. (X)

还是提高性价比吧  
全用质量好的存储设备太贵了！

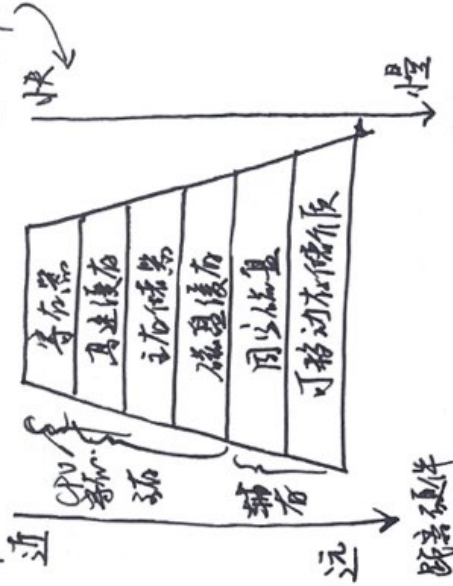
金用度是好的存储器太貴了！

彼安限制了我的想象

(५३४)

$$\frac{D/N}{1/N}$$

提高效率!



2、移信装入内存过程



绝对差入式  
可重定位差入式 (相对)  
动态二进制插入法

可重定位装入方式 (相

→ 右に  $\text{bin}$  の代入方法

基本式

分配存储空间

單一連續分配

同分區分配一分區表

动态分配—空间分区链

基于116方搜集的静态分区划分方法

10

动态可逆性分解

[illegible]

2012年12月22日

运行时的动态链接

(首次运行~1FF)

系统误差  $\sim (NF)$

最佳选择 ~ (BF)

(長环区) ~ (WF)

一、根据底图画出

12.1

音

### 3. 分页存储管理:

① 引入原因: 连续分配, "碎片", 分页离散分配。

② 数据块及地址结构

地址: 页号 + 页内偏移量

数据块结构: 页表 (页号到物理块号的映射)

③ 地址变换机构 { 基本页 ~

页号  $\rightarrow$  逻辑地址

物理地址  $\rightarrow$  物理地址

页号 + 页内偏移量

物理地址

具有快表的 ~

—— "联想寄存器"

减少 CPU 访问数据时间

④ 多级页表: 原因: 现代计算机内存较大, 导致页表占用内存空间

### 4. 分段存储管理

① 引入原因: ① 方便编程

② 基本单元与分类

③ 主要区别:

- i) 基本单元 (逻辑单位) 不同, 页/段
- ii) 大小 (页由系统决定且固定, 而段不固定)
- iii) 地址空间位置 (页: 一维线性; 段: 二维)

5. 程序存储管理方式:

好处: ① 可共享 ② 易于保护; ③ 可动态链接; ④ 减少内存碎片

① 动态增长 ② 动态链接



心得体会:

关于理论课:

1. 老师上课模式、提问+讲授+课堂自习

提问环节: 同一张图上课重点, 很多人也都有自己的理解, 加深了对知识点的认识, 同时回顾了上节课重点。

也可以更快进入学习状态

讲授环节: 生动形象, 不会PPT, 深入浅出, 通俗易懂, 有时还会讲例题, 便于深入理解知识点。

课堂自习: 课堂看书自习, 不会即问。

2. 操作系统作为计算机体系课程, 是较为抽象的。它连接了硬件和用户, 直接与计算机硬件打交道, 应该认真学习。通过长期的学习, 学习了操作系统的主要功能以及基本的操作系统编程打下了基础。

关于实验课:

我个人认为验证型实验没有什么含金量, 而且内容考套(百度一堆答案), 操作系统和IDE都比较老, 我用

Windows Server 2016 + VS 2019

建议: 改为实现一些操作系统的基本/简单的算法感觉会好点。

调了很久才通。