

发现OS考概念，考试之前随便记点。（哭）

# 操作系统的概念、功能和目标

## 概念

操作系统（Operating System, OS）是指**控制和管理**整个计算机系统的硬件和软件资源，并合理地**组织调度**计算机的工作和资源的分配，以提供给用户和其他软件方便的**接口和环境**，它是计算机系统中最基本的**系统软件**。

## 功能和目标

1. 系统资源的管理者：安全、高效
  1. 处理机管理
  2. 存储器管理
  3. 文件管理
  4. 设备管理
2. 用户和计算机硬件之间的接口
  1. 命令接口：用户直接使用
    1. 联机命令接口：说一句做一句，cmd
    2. 脱机命令接口：.bat，批处理命令接口
  2. 程序接口：用户通过程序间接调用，由一组系统调用组成.dll文件
  3. GUI
3. 实现对硬件机器的拓展（封装）

## 操作系统的基本特质

- 并发 Concurrency：系统中存在着多个运行的程序
- 共享 Sharing：资源共享，多个并发程序共同使用系统资源
  - 互斥共享
  - 同时共享
- 虚拟 Virtual：虚拟存储器技术（空分处理）、虚拟存储器技术（时分处理）
- 异步 Asynchronism：多个程序并发执行，但是由于资源优先，进程的执行不是一贯到底的，而是走走停停，以不可预知的速度向前推进。

并发和共享互为存在条件

# 计算机开机之后的工作

- BIOS: basic I/O 系统 基本输入输出系统

计算机的很多硬件中都有BIOS, 主板、显示卡、IDE控制器、网卡等中都存在BIOS

1. power on之后, CPU初始化, 之后跳到BIOS的启动代码处
2. CPU执行POST(Power On Self Test, 加电自检)
3. 查找其他设备的BIOS, 此时一般开始查找显示卡的BIOS
4. 系统BIOS开始检测CPU类型和工作频率, 测试主机内存容量
5. 系统BIOS检查标准硬件设备 包括硬盘、CD-ROM等
6. 系统BIOS检测即插即用设备, 并为这些设备分配中断DMA通道、I/O端口等资源
7. 系统BIOS更新ESCD(Extended System Configuration Data, 扩展系统配置数据)
8. 系统BIOS的启动代码将进行它的最后一项工作, 即根据用户指定的启动顺序从软盘、硬盘或光驱启动。(如果有多个操作系统, 那么也就是在这一步选择哪个操作系统被加载)

开机启动的流程图表示

## 操作系统分类

class

1. 批处理系统 batch processing sys

用户将作业交给系统管理员, 系统管理员把作业分组, 成批处理

batch 又分为

1. 单道批处理: 内存中只有一道作业
2. 多道批处理: 内存中有多道作业

2. 分时系统 time sharing

多个用户, 操作系统以时间片为单位, 轮流为每个用户服务

多个用户通过中断同时共享一台主机

3. 实时系统 real-time sys

又根据时间限制的严格程度分为两种

1. 硬hard 实时系统: 某个动作必须绝对地在规定的时刻(或规定的时间范围)发生
2. 软soft 实时系统: 能够接受偶尔违反时间规定, 并且不会引起任何永久性的损害

后来随着OS的发展, 又出现了

1. 嵌入式系统
2. 个人操作系统
3. 网络操作系统

## 操作系统的运行机制和体系结构

### 运行机制

#### 两种指令

1. 特权指令：不允许用户程序使用，一定要在处理机的核心态才能执行
2. 非特权指令：核心态和用户态均可以执行

#### 两种处理机状态

1. 内核态：此时特权、非特权指令均可执行
  1. 又称 **monitor mode, supervisor mode, system mode, privileged mode**
2. 用户态：此时CPU只能执行非特权指令

这两种处理器的状态是如何标识的呢？

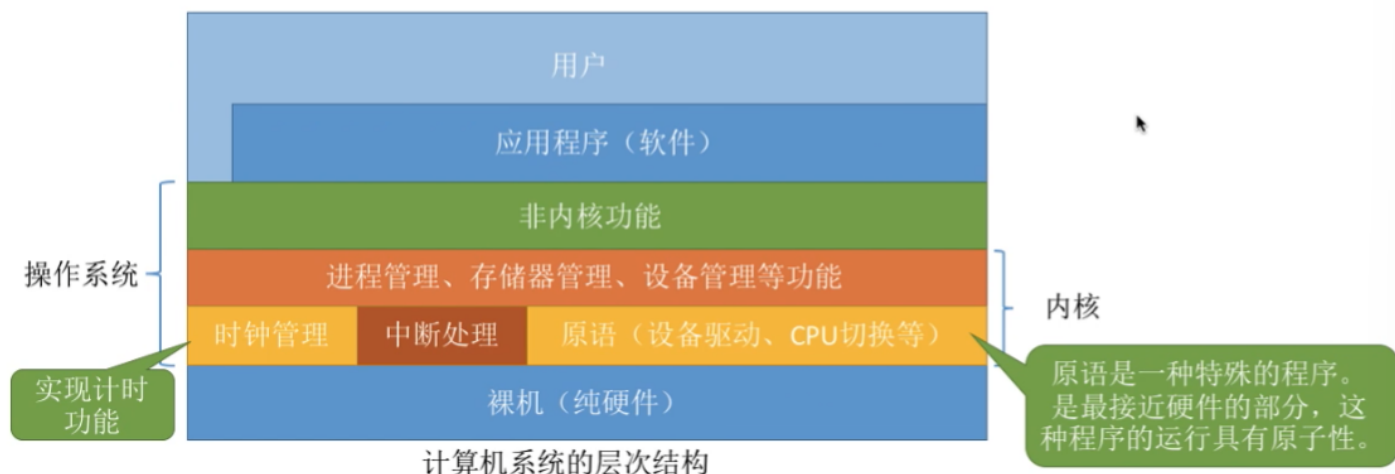
用**程序状态字寄存器PSW**中的某一位来标识。

#### 两种程序

1. 内核程序：系统的管理者，可以执行**特权指令**和**非特权指令**，运行在**核心态**
2. 应用程序：只能执行**非特权指令**，运行在**用户态**

### 操作系统内核

Yo~生活经验：我们安装完 Windows 操作系统后，会发现操作系统提供了多种多样的功能，比如“记事本”、“任务管理器”。然而，这些功能并不是必不可少的。即使没有“任务管理器”，我们仍然可以使用计算机。

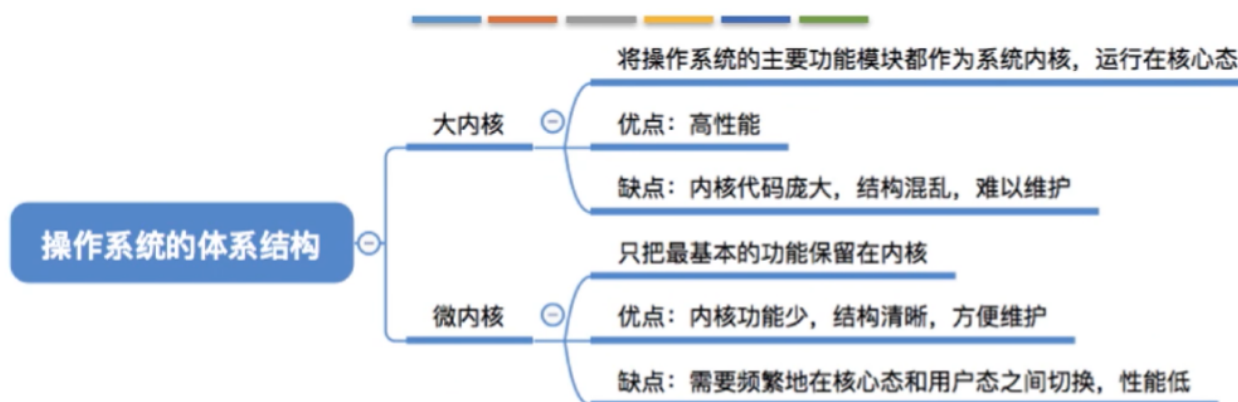


1. 时钟管理
2. 中断处理
3. **原语 (Primitives)**
4. 对系统资源进行管理的功能
  1. 进程管理
  2. 存储器管理
  3. 设备管理

## 操作系统的体系结构

1. 大内核
2. 微内核

两种内核的优缺点



类比：

操作系统的体系结构问题与企业的管理问题很相似。

**内核**就是企业的**管理层**，负责一些重要的工作。只有管理层才能执行**特权指令**，普通员工只能执行**非特权指令**。**用户态**、**核心态**之间的**切换**相当于普通员工和管理层之间的**工作交接**

**大内核**：企业初创时体量不大，管理层的人会负责大部分的事情。优点是效率高；缺点是组织结构混乱，难以维护。

## 一些PPT上的知识点

### Processor Registers

1. User-visible regs
  1. Data
  2. Addrss
    1. index
    2. segment point
    3. stack point
2. Control and status regs

1. PC
2. IR
3. PSW(program status word)
4. Condition codes or falgs:正负结果、零、溢出etc

## 指令周期

start——>fetch next -> execute -> Halt



## 指令类型

略，懒得写了

## 计算机开机

bootstrap 引导程序

## 计算机系统操作

每个device controller in charge 一类device，每个device controler 拥有local buffer

CPU moves data from/to main memory to/from local buffers

I/O指的是 从device move data 到 controller的local buffer中去

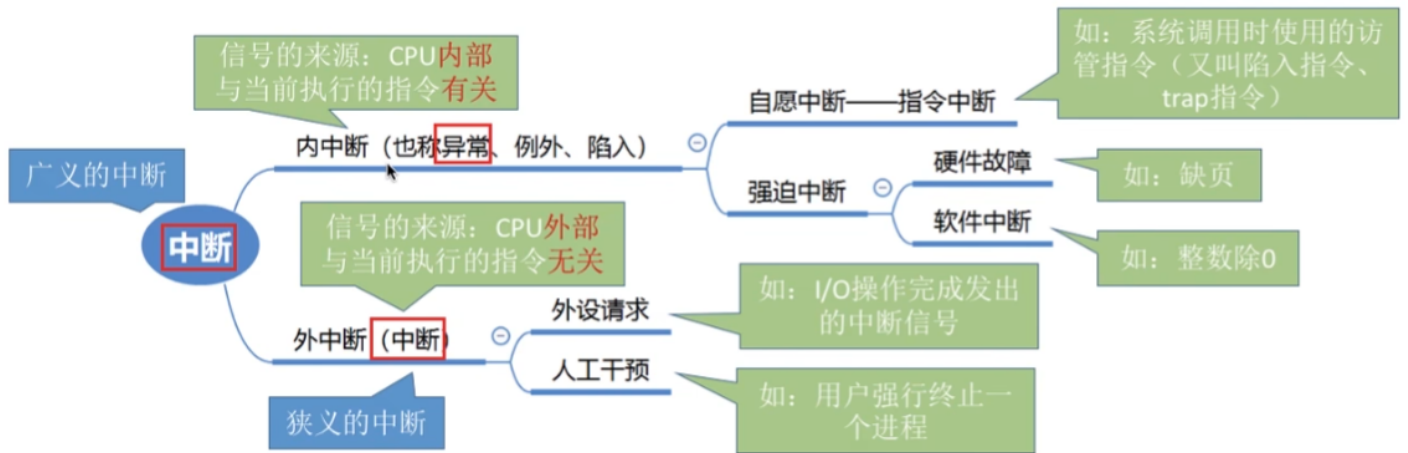
Device controler 告诉CPU它已经结束了，通过 interrupt

## 中断和异常

- CPU**收到中断信号**，对**中断处理的时候**需要切换到**核心态**
- 在I/O过程中，发出I/O请求的进程其实是在waiting（阻塞/挂起）状态的
- **中断**可以使得CPU从**用户态到核心态**，而且是**唯一**的途径
- 核心态到用户态只是需要设置一下程序状态字PSW的特殊位就可以了

中断的分类

1. program 内部出现问题，通常称为**trap**
2. 外部，通常称为**interrupt**
  1. timer
  2. I/O
  3. Hardware failure



外中断的处理过程

中断方式:

1. 轮询polling
2. 矢量中断系统 (vector interrupt system)

中断向量 (表):

contains 所有services routines (服务程序) 的开始地址

## Critical Section访问

临界区访问的三个原则

1. Mutual Exclusion: 互相排斥
2. Progress: (稳定的) 前进, 即空闲让进
3. Bounded Waiting: 有限等待, 有等待是上限

## 几种scheduler (短中长)

## CPU调度算法

FCFS

SJF

# RR

## Multilevel Queue

## Multilevel Feedback Queue

## HRRN

高响应比优先算法

非剥夺

响应比：

$$R = \frac{(W + T)}{T} = 1 + \frac{W}{T}$$

W:在ready队列的等待时间

T: CPU-burst 时间

HRRN同时考虑等待时间&CPU-burst时间

## 系统调用

- 系统调用是操作系统提供给应用程序的接口，应用程序通过系统调用来请求获得操作系统的服务
- 系统调用会使处理器从用户态进入kernel态
- 系统调用的分类（凡是和资源有关的操作、会影响到其他进程的操作，一定要OS介入，一定要通过系统调用来实现）
  - 设备管理
  - 文件管理
  - 进程控制
  - 进程通信
  - 内存管理
- 系统调用和库函数：有的库函数是对系统调用的进一步封装，比如文件读写库函数，的但不是所有的库函数都涉及系统调用（废话，求个绝对值根本用不到系统调用的嘛）
- 系统调用的过程
  - 传递系统调用参数
  - 执行trap陷入指令
  - 执行系统调用相应的服务程序
  - 返回用户程序

这里需要注意的是，**系统调用**发生在**用户态**，但是**对系统调用的处理**发生在**内核态**。

执行陷入指令会引发**内中断**，使处理器从用户态进入核心态

- trap陷入指令是唯一一个只能运行在用户态而不能运行在核心态的指令

## 进程通信

### Shared memory

1. 基于数据结构的共享
2. 基于存储区的共享

### Pipe

只能实现半双工

想要双向通信，需要设置两个管道

只有写满了才允许读，读空了才允许写。

数据一旦被读出，就没了，所以一个管道中的读进程最多只能有一个

### Message passing

消息（就像计网里面的报文）包括

1. 消息头
  1. 发送进程ID
  2. 接受进程ID
  3. 消息类型
  4. 消息长度
2. 消息体

消息传递

1. 直接通信：直接挂到消息缓冲队列
2. 间接通信：先发送到信箱

## 死锁

### 死锁发生的四个条件



1. Mutual Exclusion 互斥访问
2. Nonpreemptive 非剥夺的
3. Hold & Wait 保持并等待
4. Cycle(Circular) Waiting 循环等待

## 死锁的处理

1. 不允许死锁发生
  1. 静态策略：预防 破坏四个条件中的任意一个
  2. 动态策略：避免 **算法：银行家**
2. 允许死锁发生
  1. 死锁的检测：资源分配图
  2. 死锁消除

## Words

### **Primitives** 原语

increment 增值，加薪

halt 停止

approximation 近似值 比喻

interfere 干扰，妨碍

intermediary 中间人

suspension 暂停

external 外在的

resume 继续，恢复，重新得到

**context** 环境，上下文

preemptive 剥夺的，抢占的

nonpreemptive 非..., 非...

dispatch 派遣，调度

scheduler 调度程序

latency 等待时间，潜伏期

CPU utilization CPU利用率

throughput 吞吐量

turnaround time 周转时间

waiting time 等待时间

response time 响应时间

homogeneous 同种类的；相同类型人或事物组成的

heterogeneous 各种各样的

symmetric 对称的