

# 复习纲要

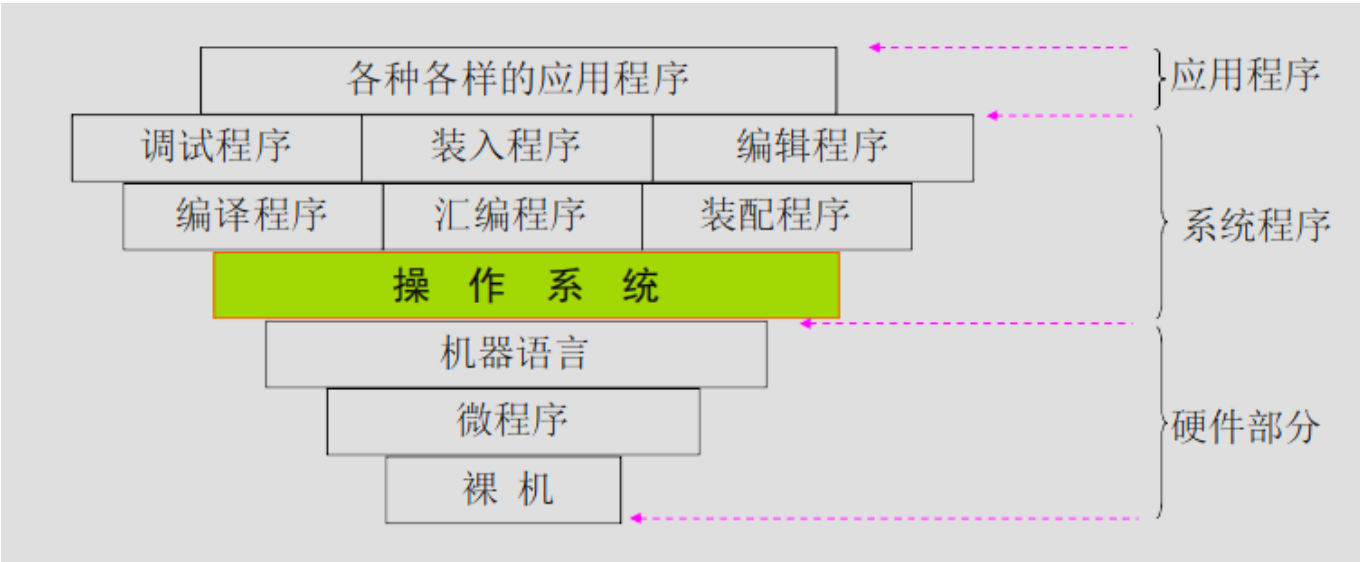
这里是 Bolaxious 的操作系统复习纲要，教材为《操作系统原理（慕课版）》（苏曙光），分九个章节：概述、硬件基础、用户界面、进程管理、死锁、进程调度、存储管理、设备管理、文件管理。

## 概要

操作系统提供了这些直观的基本功能：提供操作界面、控制程序运行、管理系统资源、配置系统参数、监控系统状态、工具软件集合。

## 定义

操作系统是一个大型的程序系统，它负责计算机系统软件/硬件资源的分配；控制和协调并发活动；提供用户接口，使用户获得良好的工作环境。简而言之就是：**管理调度资源、提供用户接口。**



上图展示了操作系统的地位。

这也就要求并赋予了操作系统以特性：具有同时处理多个任务的能力、为多个并发任务提供资源共享、具有处理随机事件(中断)的能力，简言之就是：**并发性、共享性、不确定性**

至于发展历史，我们在此略过。

## 功能

资源管理是操作系统的主要任务，从这个观点来分析操作系统的功能可以分为四个基本功能：处理机管理、存储管理、设备管理、文件管理

## 处理机管理/进程管理

我们先要了解“进程”的概念：

OS 会在任何时候和任何地点暂停或继续任何程序的运行。在并发环境下，“程序”不足以描述程序的运行过程并确保运行结果的正确，故而我们需要引入进程来确保程序的独立运行。

处理机的具体功能包括：

- 进程控制：创建，暂停，唤醒，撤销
- 进程调度：调度策略，优先级
- 进程通信：进程间通信

## 存储管理/内存管理

具体功能包括：

- 内存分配
- 内存共享
- 内存保护
- 内存扩充
- 虚拟内存

## 设备管理

设备管理的具体功能包括：

- 设备的分配和回收
- 设备的驱动机制/传输控制
- 为应用提供统一接口访问设备
  - 设备无关性
- 高效存取设备/缓冲机制

## 文件管理

文件管理为用户提供统一的文件存取接口，高效组织存储空间，提高存取效率，实现信息共享和存取控制。

我们可以这样理解：文件是设备的抽象

具体功能包括：

文件用户接口

- 存储空间管理
- 文件的操作
- 目录的操作
- 存取权限管理

## 性能

我们可以从下面几个指标来评价操作系统的性能

- 吞吐率
  - 在单位时间内处理信息的能力。
- 资源利用率
  - 设备(CPU)使用的频度
- 响应能力
  - 从接收数据到输出结果的时间间隔。
- 可移植性

- 改变硬件环境仍能正常工作的能力:代码修改量（为了适配新的硬件需要做一定的修改工作，也就是移植，如果抽象和通用的接口多，那么代码修改量就会小很多，可移植性就越高）
- 可靠性
  - 发现、诊断和恢复系统故障的能力。

## 发展历史

### 操作系统发展的四个典型阶段

- 手工操作(无操作系统) (40 年代-50 年代初)
- 单道批处理系统 ( 50 年代)
- 多道批处理系统 (60 年代初)
- 分时操作系统 (60 年代中-至今)

手工操作时期，用户依靠卡片/纸带和计算机交互，CPU 利用率低、用户独占且缺少交互；

单道批处理系统随着 IBM7094 晶体管计算机而出现，它的工作过程是管理员将多个作业输入到磁盘形成作业队列；监控程序（操作系统）依次自动处理磁盘中每个作业：装入—运行—撤出—装入—运行—撤出并循环；运行完毕，通知用户取结果，不过这种方式仍然效率较低。

多道批处理系统进行了改进：内存中存放多道程序,当某道程序因为某种原因（例如请求 I/O 时）不能继续运行时，监控程序/OS 便调度另一道程序投入运行，这样使 CPU 尽量处于忙碌状态，提高系统效率。多道技术是现代操作系统的雏形。

- 时分：分成多个时段：不同程序错开时段使用。
- 空分：分成多个单元：不同程序使用不同单元。

分时操作系统的核心概念是“时间片”（较短的时间长度），主机以很短的时间片为单位，把 CPU 轮流分配给各个终端(作业)使用，直到全部作业被运行完。Unix/Linux/Windows 都是分时操作系统。

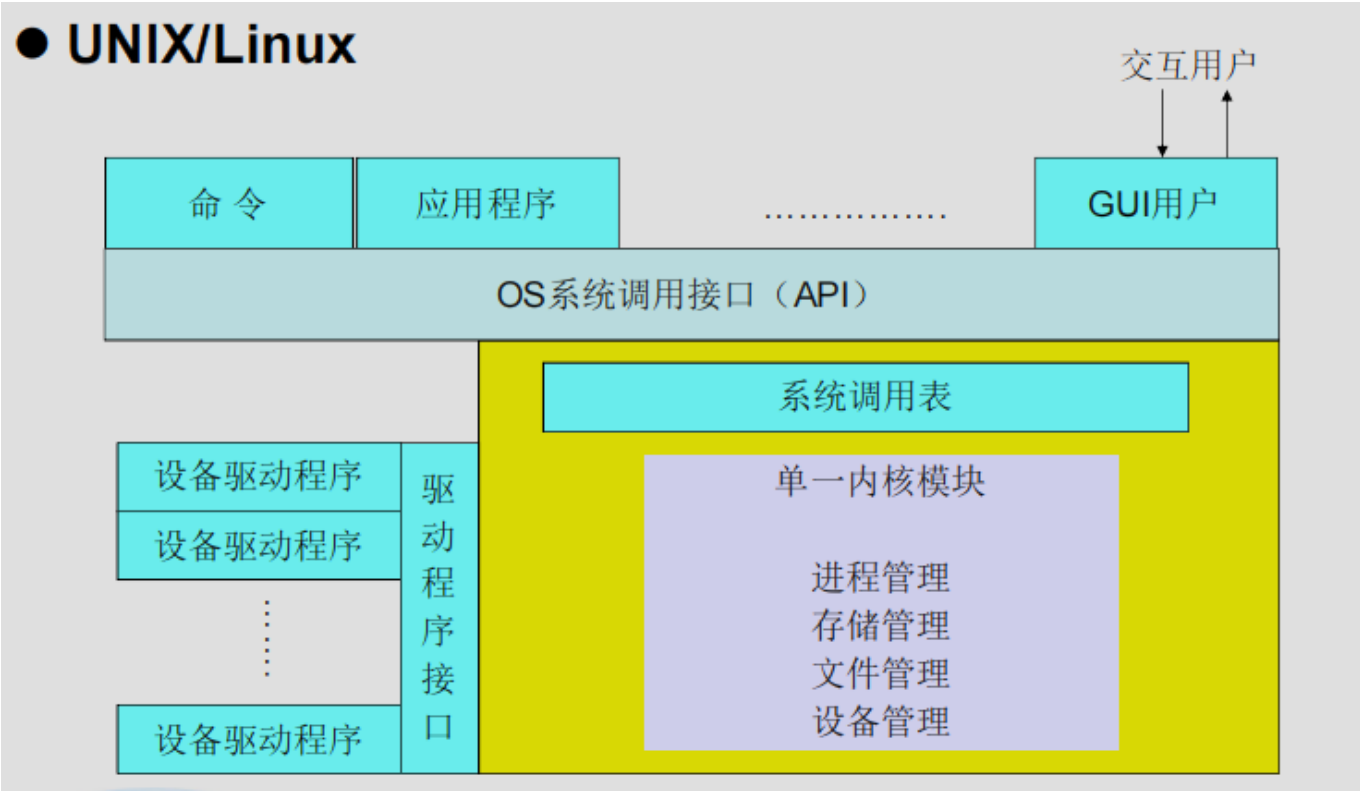
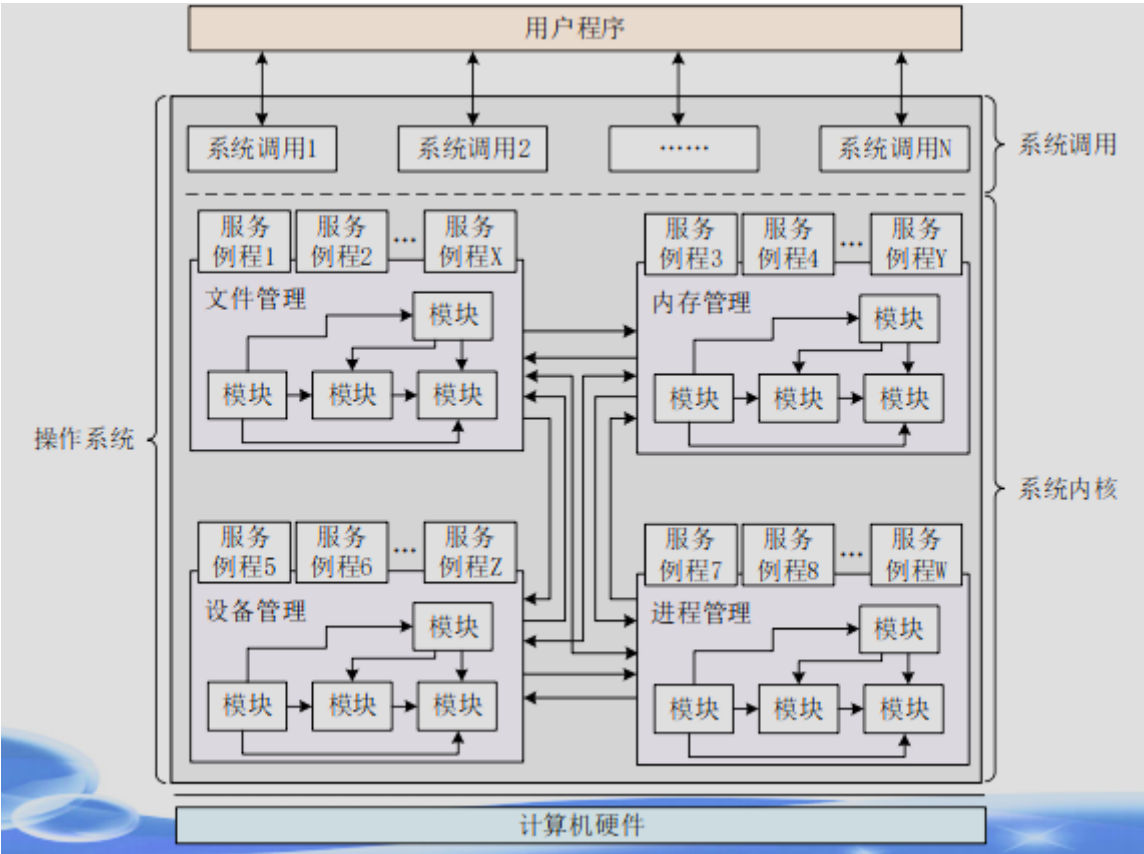
分时操作系统目前有了更多的衍化：实时操作系统/嵌入式操作系统、微机操作系统（PC 机操作系统）、多处理机操作系统、网络操作系统、分布式操作系统、嵌入式操作系统。

## 逻辑结构

逻辑结构也就是 OS 的设计和实现思路，分为：整体式结构、层次式结构、微内核结构。

### 整体式结构 OS

整体式结构以模块为基本单位构建，每个模块都有特定的功能。Unix/Linux 就是典型的整体式 OS。

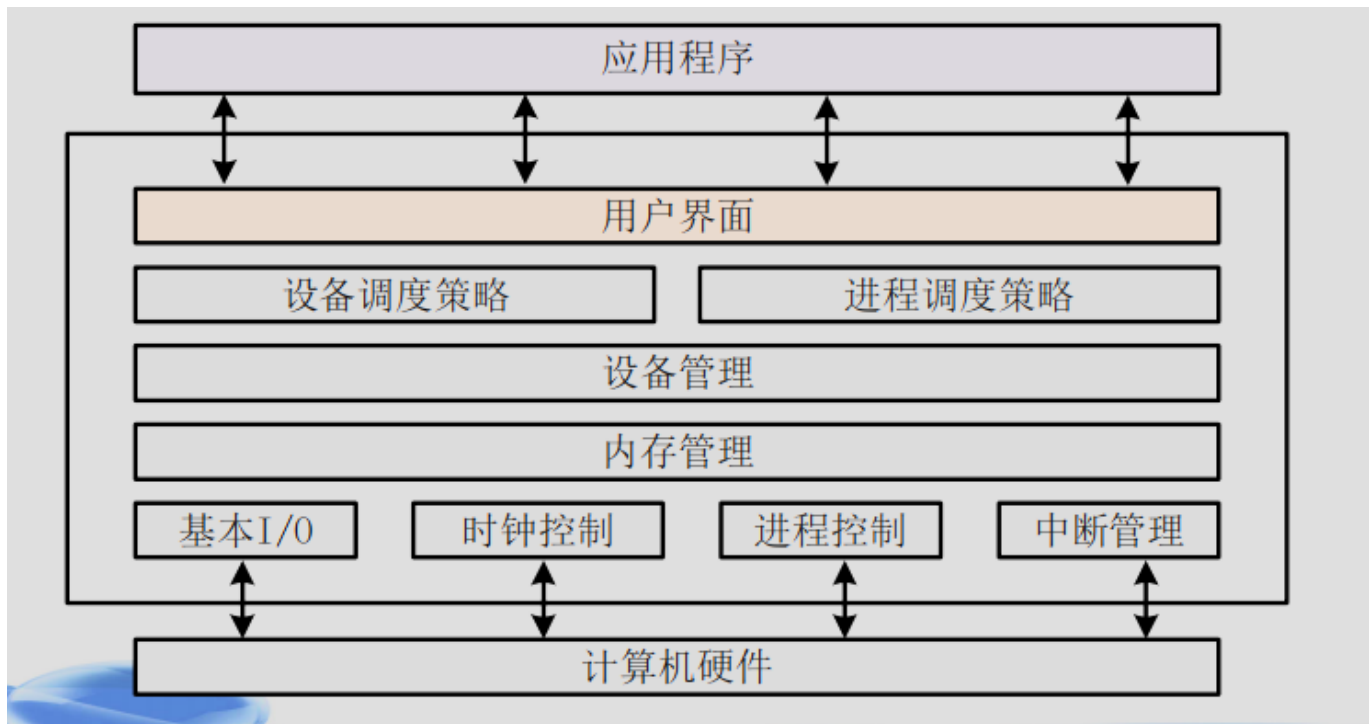


整体式结构的优点如下：

- 模块设计、编码和调试独立
- 模块之间可以自由调用

### 层次式结构

层次式结构将操作系统的功能模块按调用次序排若干层，保证各层单向依赖或单向调用。



分层原则如下：

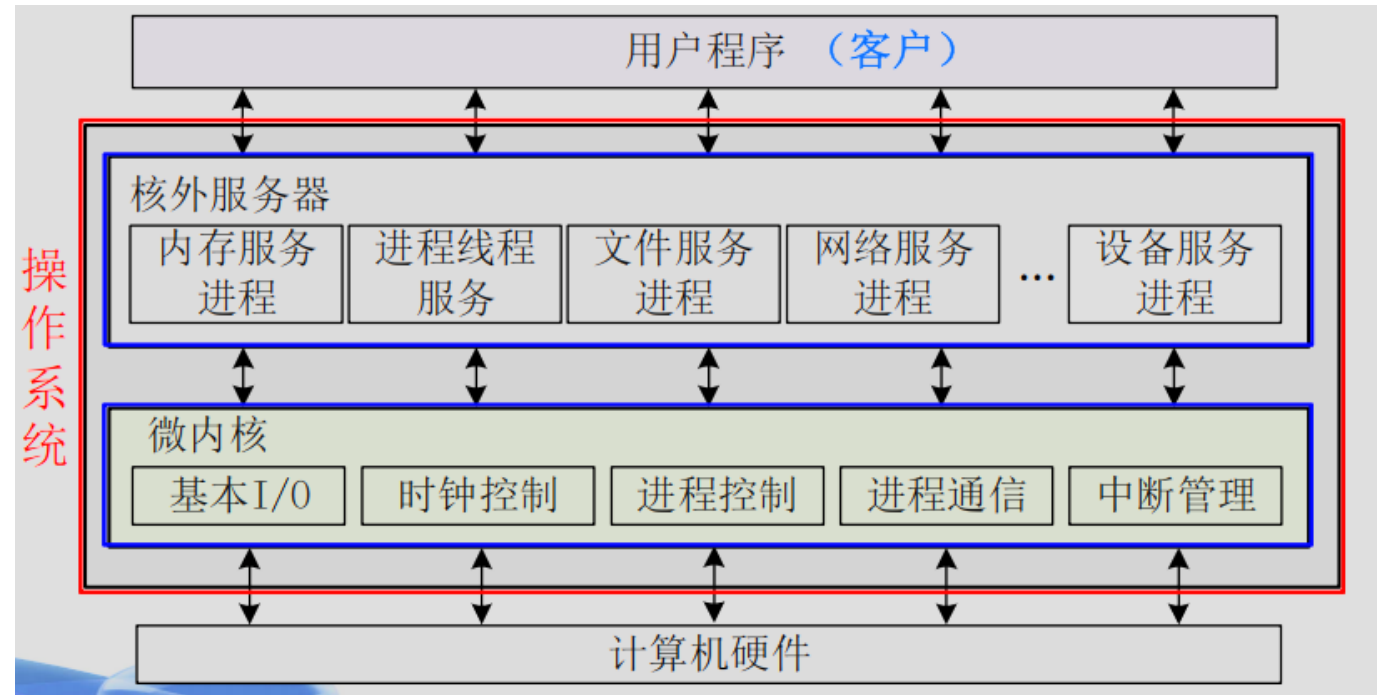
- 最底层：硬件相关
- 最顶层：用户策略/用户交互
- 中间层：按调用次序/消息传递顺序
- 较低层：共性的、活跃的服务

优点是：

- 结构清晰，避免循环调用。
- 整体问题局部化，系统的正确性容易保证。
- 有利于操作系统的移植、维护、扩充。

### 微内核结构（客户/服务器结构）

操作系统 = 微内核 + 核外服务器



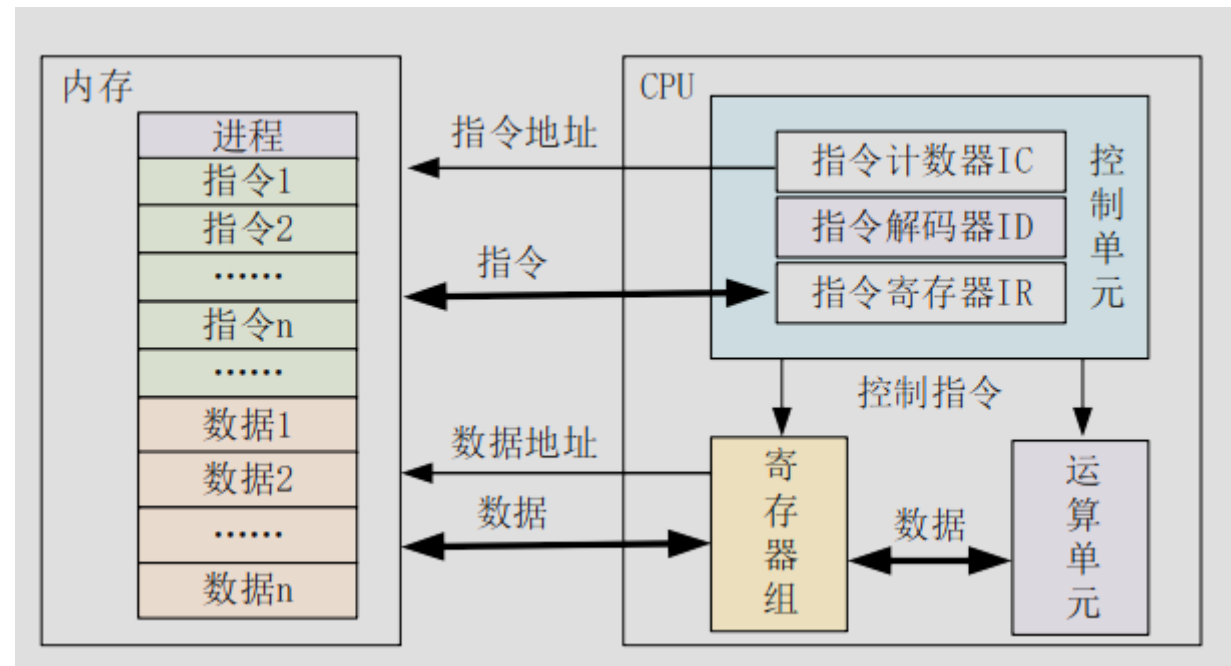
微内核体积较小，完成的是操作系统基本的核心功能和服务；  
核外服务器完成OS的绝大部分功能，等待应用程序提出服务请求。它由若干服务器或进程共同构成。

硬件基础

计算机的硬件结构主要包括：CPU、内存、外设，三者通过地址/数据/控制三条总线相连接，3条总线通过时序变换执行指令，从而实现程序的执行。  
而操作系统依赖的最基本硬件为：CPU、内存、时钟、中断。

CPU与CPU的态

CPU（Central Processing Unit）即中央处理器，它的功能主要是按照一定的逻辑流程分析和执行指令流。  
CPU再逻辑上可以划分为3个单元：分别为控制单元、运算单元和寄存器单元



为CPU设定态的目的在于为系统建立安全机制，不同的态支持程序使用不同的指令集和资源。故而我们需要了解一下操作系统需要考虑的安全问题，才能更好的理解设计态的理由。

目前OS的多道程序设计技术的特点是：多进程并发/资源共享，所以我们需要考虑到：

- 防止进程的信息被非法访问
- 防止进程随意存取系统资源
- 防止进程修改系统安全机制

提出的解决策略为：

- 对于软件，分级/分类（例如分为可信软件和不可信软件）
  - 对于可信软件
    - 可以修改安全保护机制
    - 可以存取系统资源
    - 拥有普通指令集 + 特权指令集
- 对于硬件，分模式（设置访问屏障）
  - 模式（态）：描述指令使用和资源访问的权限

Intel x86架构的CPU支持 Ring0~3 的特权级，权限逐次降低，Unix/Linux/Windows 只使用了0和3两种特权级。

对于CPU的态，分类为3种：

- 核态(Kernel mode)
  - 能够访问所有资源和执行所有指令
  - 管理程序/OS内核
- 用户态 (User mode, 目态)
  - 仅能访问部分资源，其它资源受限
- 管态(Supervisor mode)，它介于核态和用户态之间。

通过设置模式位表示态，CPU就能通过它来进行权限判断，从而保证系统安全。

用户态和核态之间的转换：

- 用户态向核态转换
  - 用户请求OS提供服务
  - 用户进程产生错误（内部中断）
  - 用户态企图执行特权指令
  - 发生中断
- 核态向用户态转换的情形
  - 一般是中断返回：IRET

特权指令包括：

- 涉及外部设备的输入/输出指令
- 修改特殊寄存器的指令
- 改变机器状态的指令

## 内存

内存也叫主存，分为只读存储器-ROM、随机存储器-RAM和高速缓存存储器-Cache三种。

ROM主要存BIOS之类的系统程序（因为它不可读）

RAM是内存条。

Cache位于CPU和内存之间，用来缓存内存中的某一小部分数据。

## 时钟

计算机设置有一个硬件定时器，能够定时向处理器发出时间信号，帮助处理一些依赖时间概念的地方。

## 中断机制

中断是指CPU对突发的外部事件的反应过程或机制，CPU收到外部信号（中断信号）后，停止当前工作，转去处理该外部事件，处理完毕后回到原来工作的中断处（断点）继续原来的工作。

中断包括一些概念如：中断源、中断类型、断点、现场、现场处理，不过记不住，不写了。

中断响应过程如下：

- 识别中断源
- 保护断点
- 保护现场(中断服务程序的前段)
- 中断服务程序(主体)
- 恢复现场(中断服务程序的末段)
- 中断返回(恢复断点)

引入中断机制目的在于实现并发、实现实时处理、和自动处理故障。

## BIOS

BIOS（基本输入输出系统）是被固化到计算机中的一组程序，提供最基础最直接的硬件操控，它的主要功能有以下四个：

- 加电自检和初始化
- 设置CMOS参数
- 基本输入输出服务
- 加载并引导操作系统启动

### 加电自检和初始化

加电自检（Power On Self-Test, POST）初始化基本硬件（CPU、内存、显卡），自检若由严重错误则停机、不严重故障则给出提示。

### CMOS设置

计算机加电引导过程中通过特殊热键进入到CMOS设置程序中，设置基本的系统参数：系统日期、时间、系统口令等等。

### 基本输入输出服务

BIOS主要是为应用程序、操作系统提供硬件接口服务，例如显示字符串、读写磁盘等等。



基本输入输出处理程序是通过中断服务指令的形式来实现的，每组服务都有一个专门的中断号，BIOS使用的中断类型为10H ~ 1FH，每组服务又分为不同子功能，用子功能编号标识。

使用案例：使用INT 13H软盘I/O类服务中的02H子功能读取软盘第21个扇区的内容到内存1000h: 0000h处

● 读第21号扇区到内存1000H处

■ 注：假定每个磁道有18个扇区

1

MOV

AX,

1000H

2

MOV

ES,

AX

3

MOV

BX,

0

; 内存目标地址

4

MOV

AL,

1

; 扇区数

5

MOV

CH,

0

; 柱面号 (C)

6

MOV

DH,

1

; 磁头号 (H)

7

MOV

CL,

3

; 扇区号 (S)

8

MOV

DL,

0

; 驱动器号

9

MOV

AH,

02H

; 子功能号

10

INT

13H

; BIOS中断号

加载并引导操作系统启动

● 现场引导方式

■ OS文件存储在本地存储设备

加电

ROM/BIOS

CMOS

POST

启动程序

RAM

引导程序

OS内核

OS主体

引导设备/硬盘

引导程序

OS内核

OS主体

● 下载引导方式

■ OS文件存储在网络存储设备中

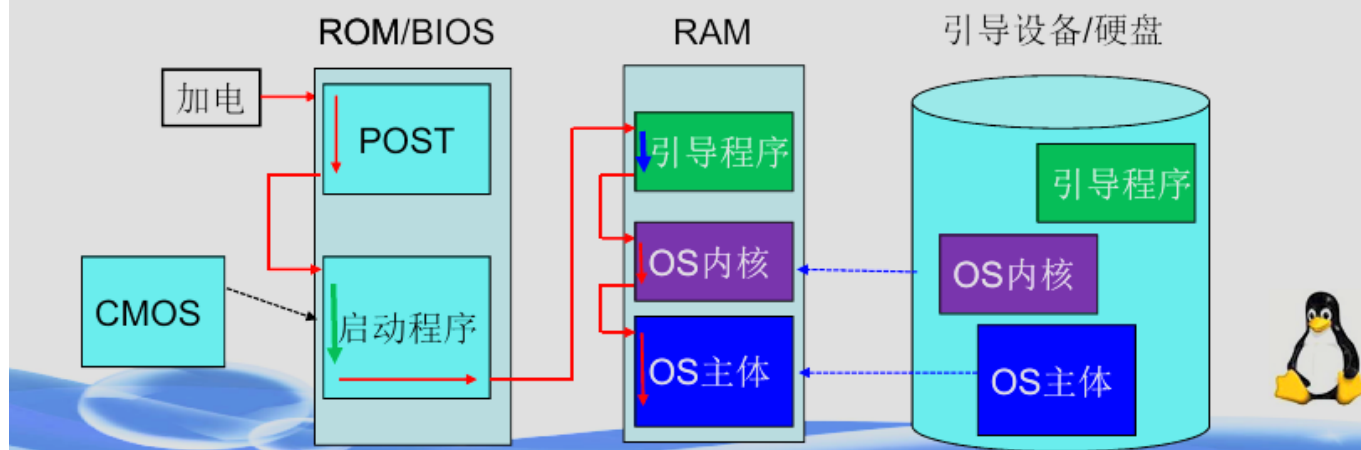
操作系统的启动过程

从加电到用户工作环境准备好的过程分为三个阶段：初始引导、核心初始化、系统初始化，它们做的工作分别为：把OS内核装入内存并使之开始工作接管计算机系统、OS内核初始化系统的核心数据、为用户使用系统作准备，使系统处于待命状态

# Linux的启动过程

## ● LINUX的启动过程

- POST → 启动程序 → MBR → KERNEL映像 →
- KERNEL映像边自解压并边执行 →
- 内核初始化 → 内核启动 →
- init进程 →



这里简略了Linux的启动过程，包括MBR、GRUB的一些概念。

## 计算机虚拟化/虚拟机

PPT上没有，略过。

## 操作系统的生成

指根据硬件环境和用户需要，配置和构建操作系统的过程。

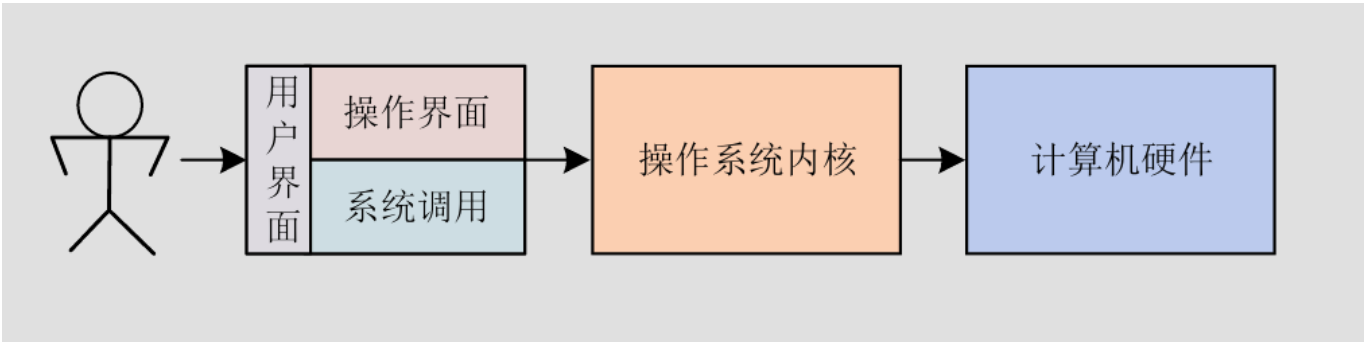
这里包括一个[Linux编译内核并安装内核的实验](#)

## 用户界面

用户环境是指计算机用户工作的软件环境，包括命令行环境、桌面环境、相关的用户使用手册。

## 用户界面概念

操作系统提供给用户控制计算机的机制(用户接口)，可分为操作界面（操作命令、批处理命令、图形用户界面）和系统调用（是提供给程序员在编程时使用的接口，取得操作系统服务的唯一途径）



操作界面

分为：

- 图形用户接口/GUI
- 操作命令（普通命令）
- 批处理与脚本程序(bat、shell)

命令	作用	命令	作用
ls	列举子目录和文件	find	查找文件
ps	列举进程	whereis	查找文件目录
top	列举进程	man	查看命令帮助信息
echo	输出字符串	cp	拷贝
cat	读文件的内容	inode	查看文件节点
cd	改变目录	tar	压缩和解压
chmod	改变文件属性	rm	删除文件和文件夹
mount	挂载文件系统	umount	卸载文件系统
insmod	安装模块	rmmod	卸载模块

Bash:[https://blog.csdn.net/qq\\_26690505/article/details/109361345](https://blog.csdn.net/qq_26690505/article/details/109361345)

这里省略了重定向和管道命令，它们是特殊的操作命令，可以结合常规的操作命令来实现一些特殊的效果和功能。

系统调用

系统调用(System Call, System Service Call)是操作系统内核为应用程序提供的服务/函数。例如：printf、exit、INT 21H等等

## ■ 特点

- ◆ 内核实现
- ◆ 存取核心资源或硬件
- ◆ 调用过程产生中断
  - 用户态 ↔ 核态
  - 自愿中断

中断

