



# 1.3 操作系统结构和运行模型

1.3.1 操作系统结构设计

1.3.2 操作系统内核

1.3.3 操作系统结构分类

1.3.3 操作系统运行模型

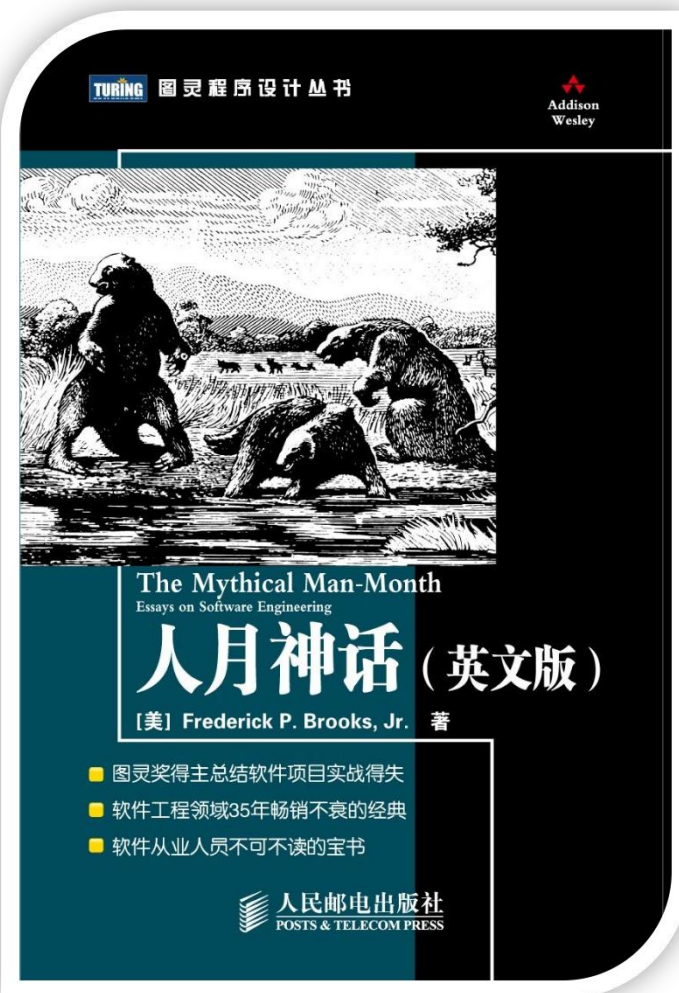


## 1.3 操作系统结构和运行模型

- 操作系统设计呈现出以下特征：

1. 复杂程度高
2. 生成周期长
3. 正确性难保证

- ◆ Windows 2000 超过**3200万行**语句，有2500个主要开发人员参与开发
- ◆ Linux 内核 拥有3.7万个文件，**1500万行**语句，主要参与人员达1300人（截至2011年）
- ◆ Android 4.0 包含**100万行**语句（截至2011年）





## 1.3 操作系统结构和运行模型

- 一个操作系统即使开发完成，仍然是无生命的，必须开发该系统下运行的大量的应用程序；
- 待应用程序开发问世后，用户还必须通过文件、培训及实践去学会操作和使用
- 这意味着用户拥有并使用的是10年或20年前的操作系统技术
- 而计算机硬件却在不断地更新换代，刚刚开发完成的操作系统又需要升级



# 1.3.1 操作系统结构设计

- 操作系统结构设计有三层含义

1. 研究操作系统的整体结构 (1.4.1)

- 功能如何分块、模块间如何交互、构造的过程和方法

2. 研究操作系统程序的局部结构 (1.4.3)

- 数据结构、控制结构

3. 研究操作系统运行模型 (1.4.4)

- 操作系统服务例程与应用进程如何交互、运行



## 1.3.2 操作系统内核

- **内核**不是进程，而是支持系统运行基本功能的一组程序模块。

内核对硬件处理器及有关资源进行首次改造，以便给进程的  
执行提供良好的运行环境

- 内核分为：

- **微内核**：内核很小，仅具有极少的必须功能，其它功能都在核外实现。通过微内核提供的消息传递机制完成其余功能模块间的联系。  
内核和核外服务程序的开发是分离的
- **单内核**：内核具有较多的功能，运行时是一个大的二进制映像，模块间的联系通过函数或过程调用实现



## 1.3.2 操作系统内核

- 内核的功能

- 中断处理：

- 为了缩短屏蔽中断的时间，增加系统内的并发性，内核通常仅进行有限的、简短的处理：分析中断事件的类型和性质，进行必要的状态修改，然后交给内核之外的（可中断）进程去处理

- 时钟管理：时间片调度，定时唤醒等

- 短程调度：主要职能是分配处理器，处理器转让，保护恢复现场等

- 原语管理：

- 原语是内核中实现某一功能的不可中断过程
    - 原语由内核完成，系统调用由系统进程实现



## 1.3.2 操作系统内核

- 内核的属性

1. 由中断驱动

- 只有当发生中断事件后由硬件交换程序状态字才引出操作系统的内核进行中断处理，且在处理完中断事件后内核自行退出

2. 内核的执行是连续的

- 在内核执行期间不能插入[内核以外的程序](#)执行
- 可陷入到高级中断

3. 可以在屏蔽中断状态下执行

- 在处理某个中断时，为避免中断的嵌套可能引起的错误，必须屏蔽该级中断

4. 可以使用特权指令

- 现代计算机都提供常态和特态等多种机器工作状态，有一类指令称为特权指令，只允许在特态下使用，规定这类指令只允许内核使用，可防止系统出现混乱



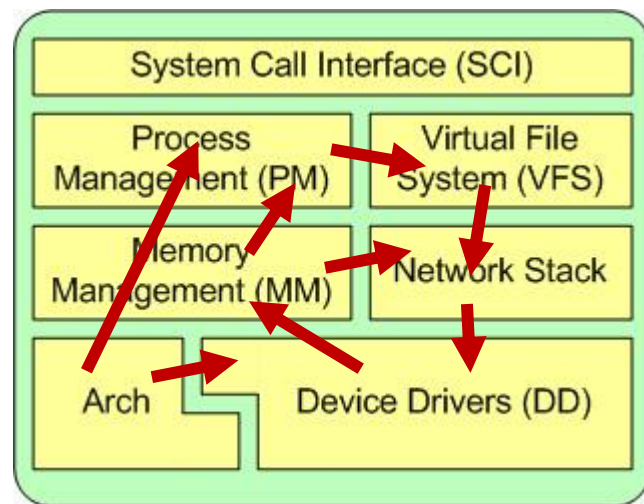
## 1.3.3 操作系统结构分类

### • 单体式结构

- 以功能模块为操作系统的基本单位
  - 每个模块有一定的独立功能，若干模块可协作完成某个功能
  - 模块间调用方法自由、随意，可不加控制
  - 模块分别设计、编码、调试

- 结构紧密
- 组合方便
- 灵活性高
- 系统效率高

- 模块独立性差
- 模块间调用关系复杂
- 系统结构不清晰
- 难以保证系统正确性







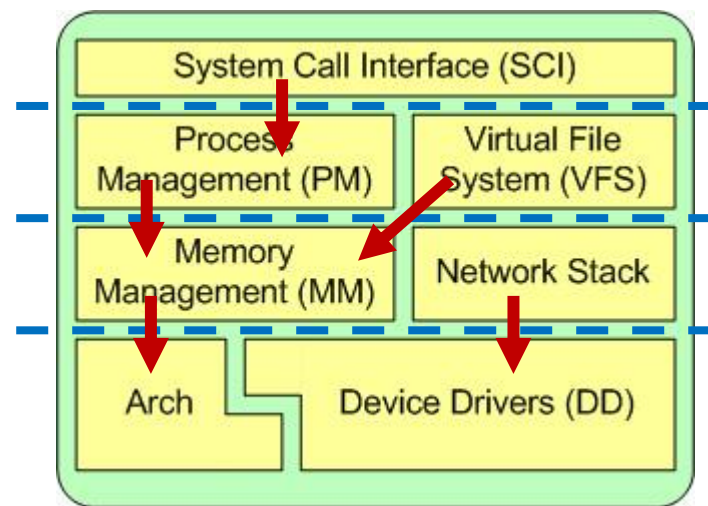
## 1.3.3 操作系统结构分类

### • 层次式结构

- 把操作系统划分为内核和模块（进程）
- 模块（进程）按功能调用次序排列成层次
  - 层次间只能单向依赖和调用，高层可调用低层功能，反之不能
  - 层次间不构成循环调用关系

- 规范层次依赖和调用关系
- 系统接口少、简单
- 逐层保证正确性

- 层次间通信开销较大
- 影响系统效率





## 1.3.3 操作系统结构分类

- 层次式结构构造方法

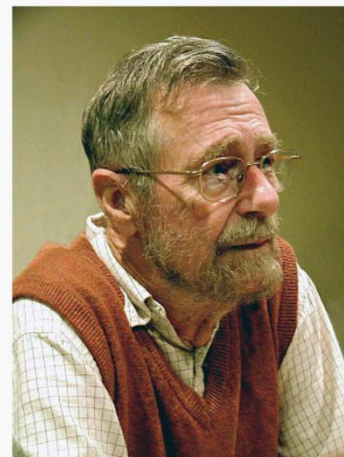
- 自底向上方法

- 从裸机开始，逐步添加各层软件，形成接近目标虚拟机的系统

- 自顶向下方法

- 自顶向下方法从目标系统出发，通过若干层软件过渡到宿主机器，其实质是对目标系统的逐步求精

艾兹格·戴斯绰  
Edsger Wybe Dijkstra



出生	1930年05月11日 荷兰鹿特丹市
逝世	2002年8月6日（72岁） 荷兰Nuenen市

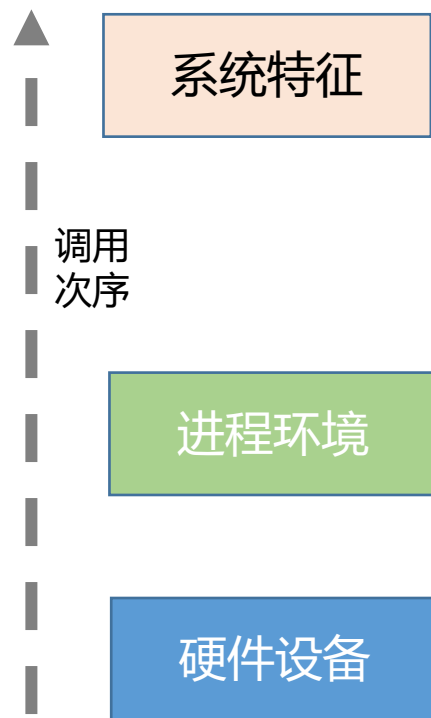
THE 多道程序设计系统，1968



## 1.3.3 操作系统结构分类

### • 层次式结构分层原则

- 把与机器硬件有关的程序模块放在最底层，以便把硬件与其它层隔离开来
- 为进程（线程）的正常运行创造环境和提供条件的内核程序应该尽可能放在底层
- 反映系统外特性的软件放在最外层
- 按照实现操作系统命令时模块间的调用次序或按进程间单向发送信息的顺序来分层



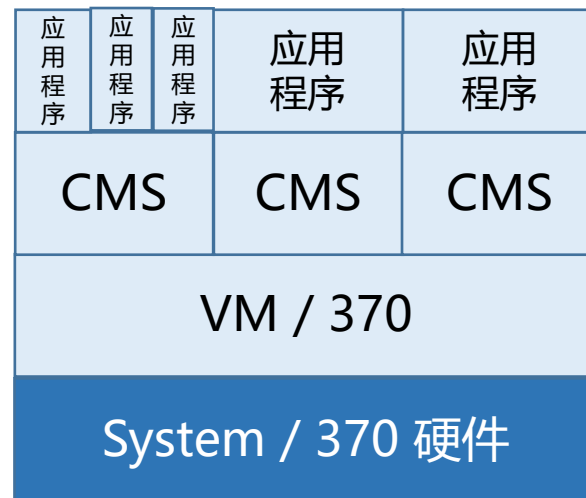


# 1.3.3 操作系统结构分类

## • 虚拟机结构

### • 设计目标:

- 使分时系统具有以下两个特性:
  - 多道程序
  - 一个具有比裸机更方便、界面扩展的计算机
- 将此二者彻底地隔离开来



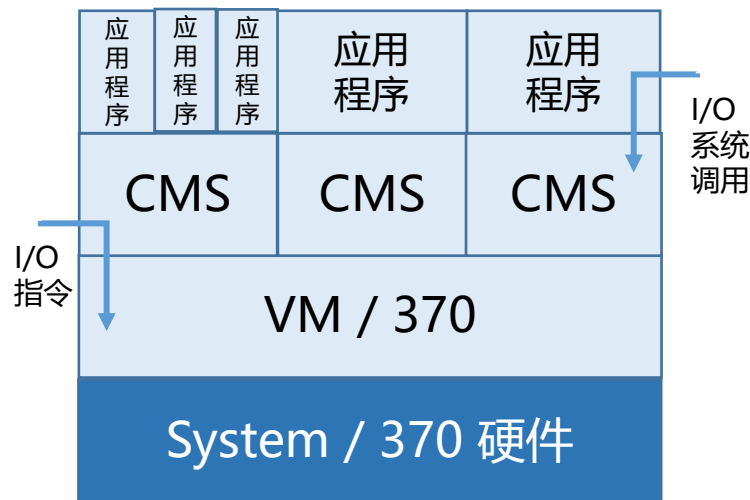
CMS: Conversational Monitor System



# 1.3.3 操作系统结构分类

## • 虚拟机结构

- 物理计算机资源通过多重化和共享技术可改变成多个虚拟机
- 通过用一类物理设备来模拟另一类物理设备，或通过分时地使用一类物理设备，把一个物理实体改变成若干个逻辑上的对应物
- 物理实体是实际存在的，而逻辑上的对应物是虚幻的、感觉上的



CMS: Conversational Monitor System



# 1.3.3 操作系统结构分类

## • 虚拟机结构

- 将一台物理计算机虚拟成了若干台逻辑计算机，而且每个虚拟计算机可以运行不同的操作系统，每个操作系统上又可并发运行多道程序
- 进程其实并不直接运行在虚拟计算机上，而是运行在虚拟计算机上运行的操作系统上，从而不同进程赖以运行的内核（平台）可以不相同

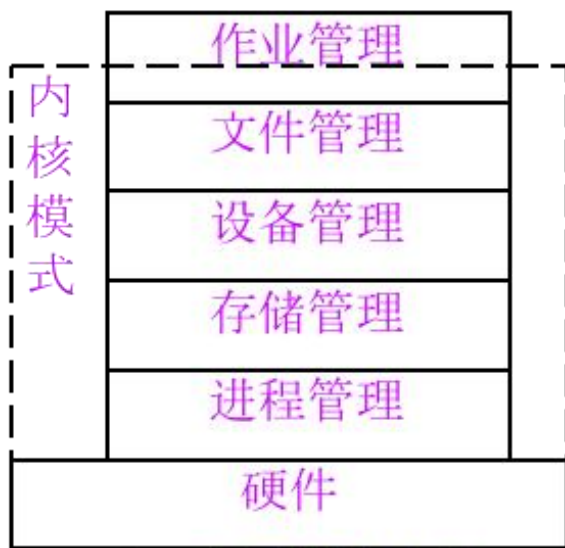
应用程序	应用程序	应用程序	应用程序	应用程序
虚拟OS			虚拟OS	虚拟OS
操作系统				
物理硬件				



## 1.3.3 操作系统结构分类

### • 微内核结构

- 封装所有应用必需的核心功能，形成微内核（microkernel）
- 其它功能形成运行在用户态的服务进程。



1) 分层结构内核



2) 微内核



# 1.3.3 操作系统结构分类

- 微内核结构

- 将操作系统分成两大部分

1. 运行在用户态并以客户/服务器方式活动的进程
2. 运行在核心态的内核

- 除内核部分外，操作系统的其他部分被分成若干个相对独立的进程，每一个进程实现一类服务，称服务器进程





## 1.3.3 操作系统结构分类

- 微内核结构

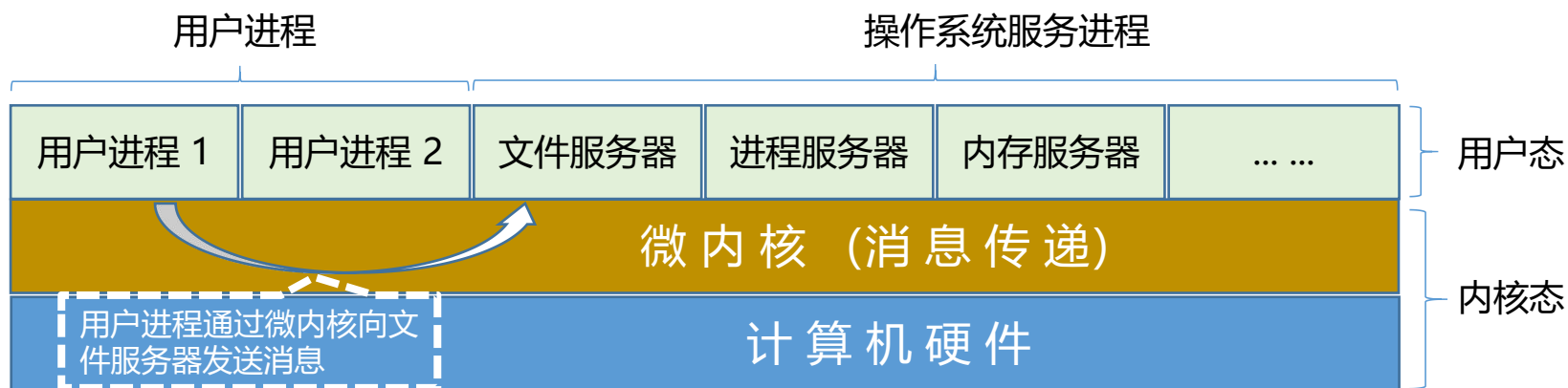
- 用户进程也在用户态，以客户/服务器方式活动
- 每个进程具有不同的虚拟地址空间，客户和服务进程之间采用消息传送进行通信，而内核被映射到所有进程的虚拟地址空间内，它就可以控制所有进程



# 1.3.3 操作系统结构分类

## • 微内核结构

- 客户进程发出消息，内核将消息传送给服务器进程，服务器进程执行客户提出的服务请求，并通过内核发送消息将结果返回给客户





## 1.3.3 操作系统结构分类

- 微内核结构

- 微内核只实现极少任务，特别是那些直接依赖于硬件的功能
  - 进程通信
  - 少量内存管理
  - I/O和中断管理
- 主要起到信息验证、消息交换的作用
- 这种结构也被称为客户/服务器与微内核结构



# 1.3.3 操作系统结构分类

## • 微内核结构

- 微内核只实现极少任务，特别是那些直接依赖于硬件的功能
  - 进程通信
  - 少量内存管理
  - I/O和中断管理

例：对存储管理而言，为了实现进程级的保护，微内核必须控制地址空间的硬件设施，因此：

- 内核负责把每个虚页面映射到物理页框
- 大量的存储管理功能，包括进程地址空间之间的相互保护、页面淘汰算法等功能可在内核外实现

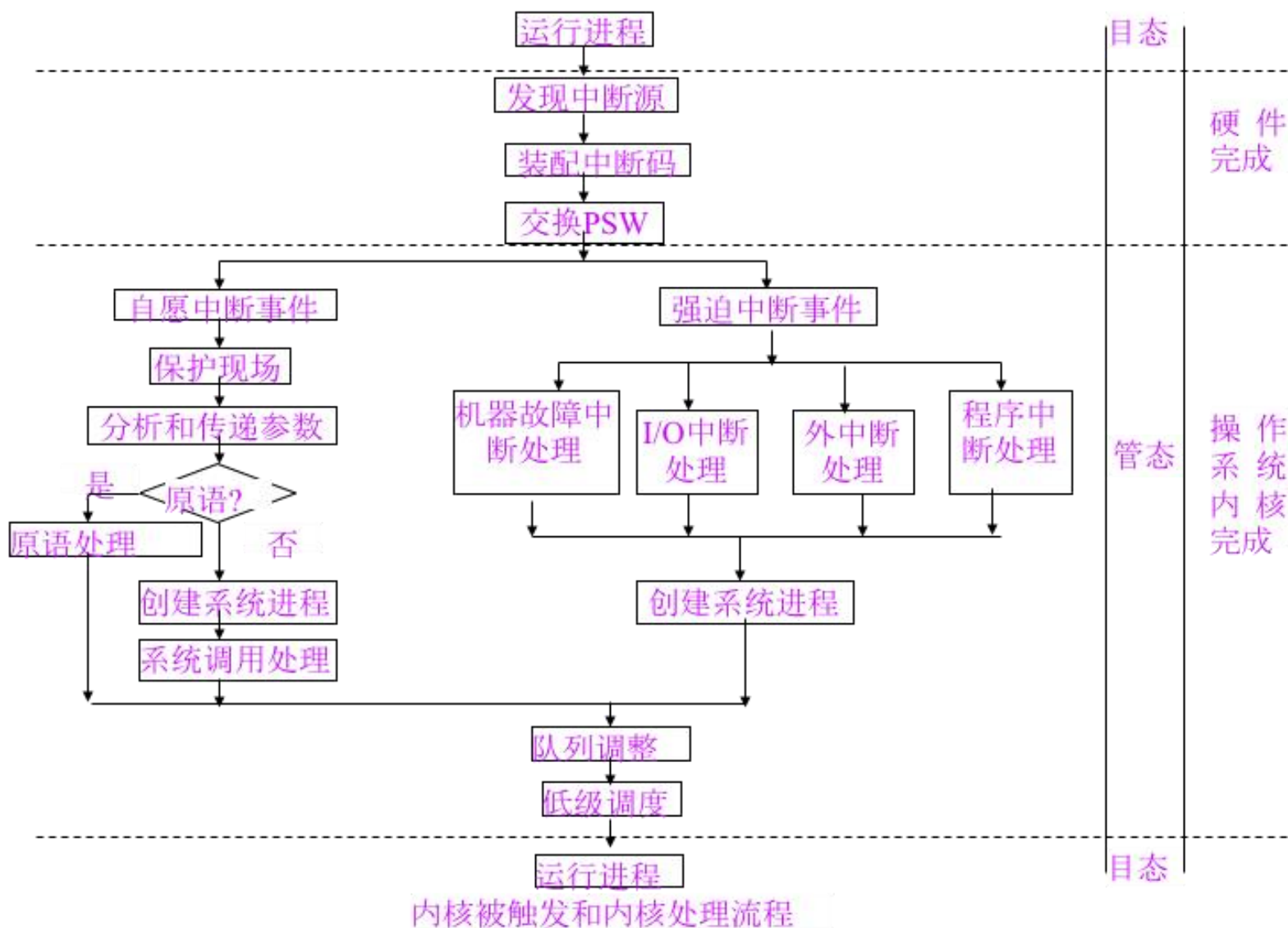


## 1.3.3 操作系统结构分类

### • 微内核结构

- 提供一致性接口，不用区分内核级服务还是用户级服务
- 良好的可扩充性，增加新功能只需更新服务器即可
- 可移植性好，与CPU相关的代码都在微内核中
- 支持分布式系统，只需要发布消息，不必知道服务器是在哪台硬件设备上

- 所有进程通信，都必须通过内核的通信机制才能进行通信
- 系统效率低





# 回顾

- 1、操作系统通过哪两种方式向用户提供服务？
- 2、什么叫程序接口？
- 3、什么叫操作接口？
- 4、系统调用与机器指令的区别？
- 5、系统调用的参数传递方式？
- 6、系统调用与函数调用的区别？
- 7、有哪两类作业级接口？
- 8、什么叫命令解释程序？
- 9、什么叫系统程序？
- 10、操作系统设计呈现出特征？
- 11、微内核和单内核？



# 回顾

- 12、操作系统的整体结构
- 13、操作系统的局部结构
- 14、操作系统的运行模型
- 15、单体式结构？缺点和优点
- 16、层次式结构？缺点和优点
- 17、虚拟机结构？缺点和优点
- 18、微内核结构？缺点和优点





## 1.3.4 操作系统运行模型

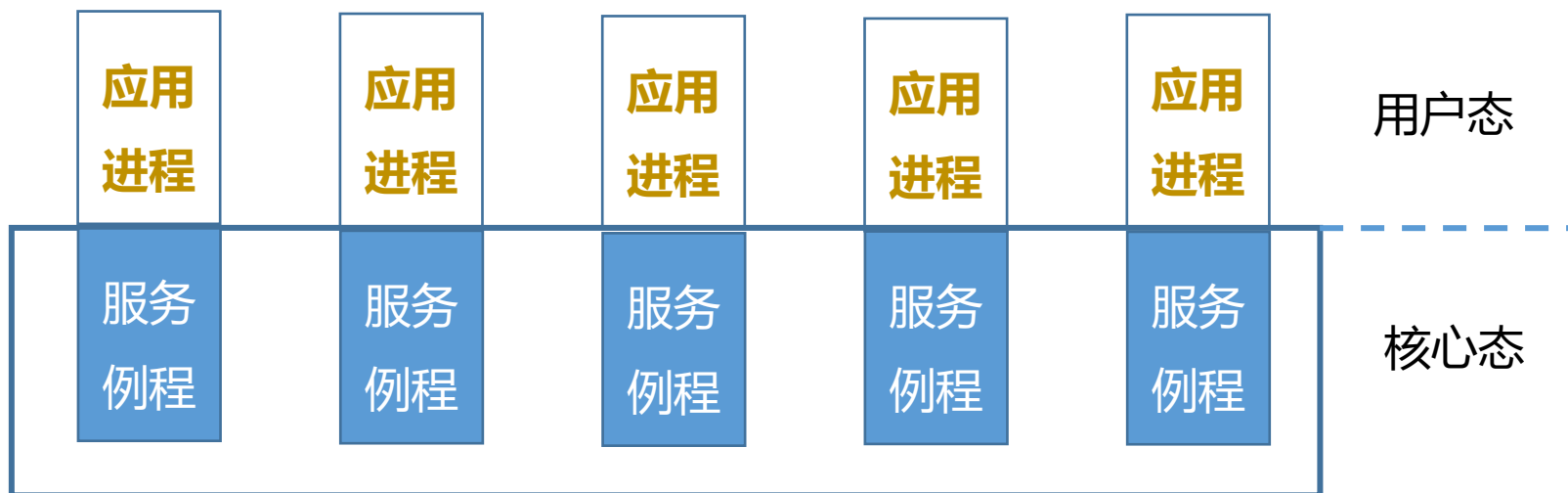
- 操作系统服务例程与应用进程如何交互、运行
  - 两个问题：
    1. 操作系统组成子程序为用户调用，还是组成进程提供服务
    2. 操作系统程序运行在系统空间，还是用户空间
  - 两种模型：
    1. 服务例程嵌入到应用进程中运行
    2. 服务例程作为独立进程运行



## 1.3.4 操作系统运行模型

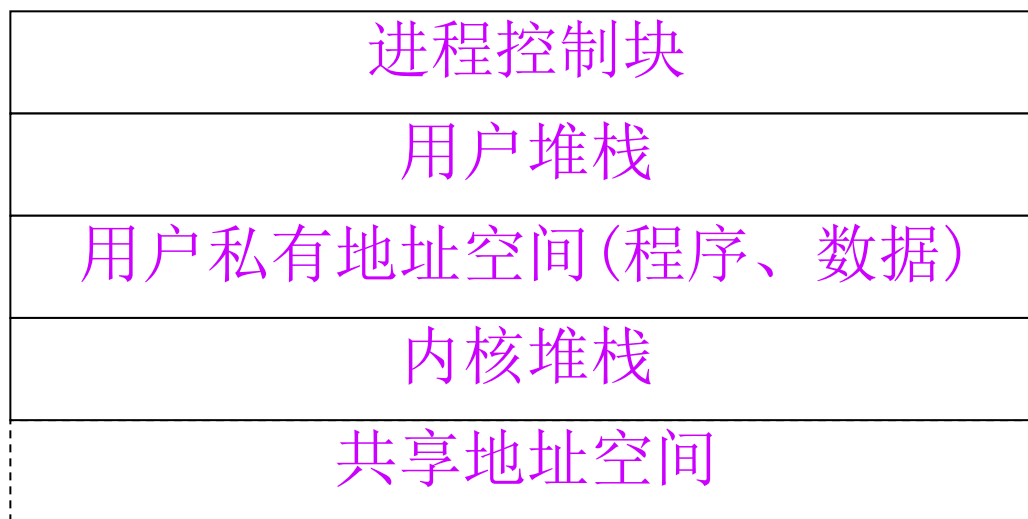
### 1. 服务例程嵌入到应用进程中运行

- 操作系统的地址空间被包含在用户进程的地址空间中
- 操作系统例行程序也在用户进程的上下文环境中执行
- 只切换处理器状态，而不切换进程，来提高效率





## 1.3.4 操作系统运行模型



服务例程嵌入应用进程的  
执行模型下的  
进程映像



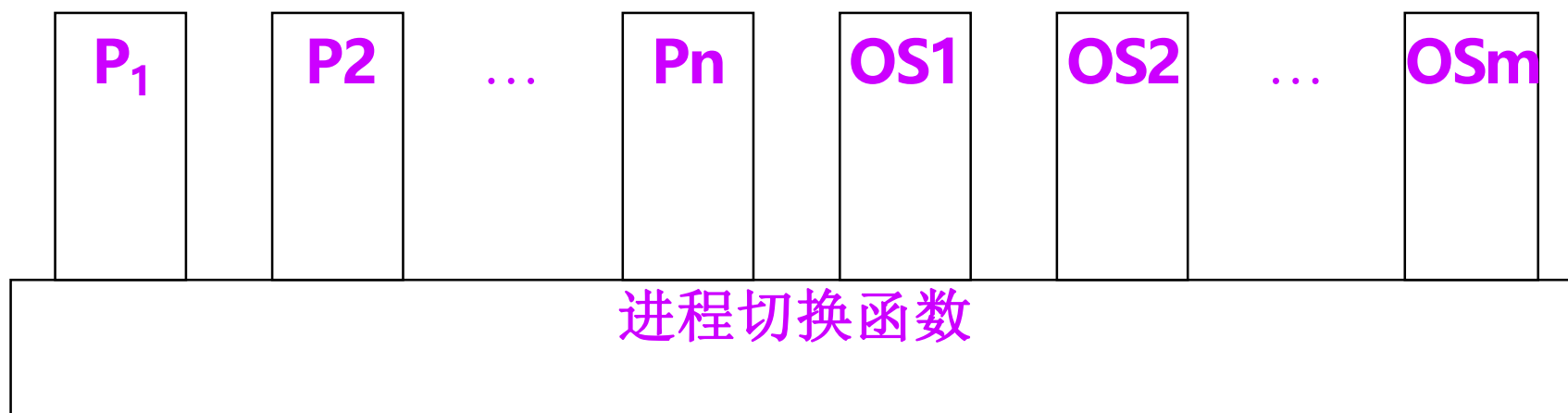
# 1.3.4 操作系统运行模型

## 2. 服务例程作为独立进程运行

- OS功能(函数)作为进程执行的模型把操作系统组织成一组系统进程
- 即操作系统功能是这些系统进程集合运行的结果，这些系统进程也称为服务器或服务进程
- 与用户进程或其他服务器进程之间构成了客户/服务器关系



## 1.3.4 操作系统运行模型



**OS**功能作为独立进程执行的模型



## 1.3.4 操作系统运行模型

- 服务例程作为独立进程运行的优点：
  - 首先，它采用了模块化的操作系统实现方法
  - 其次，操作系统功能被组织成独立的进程，有利于操作系统的实现、配置和扩充
  - 最后，这一结构在多处理器多计算机的环境下非常有效，一些操作系统服务可指派到专门处理器上执行



# 第一章作业

一、线上作业 <http://hhu.fanya.chaoxing.com/portal>

- 1、课堂在线任务点1要看完
- 2、课堂在线第一章线上作业

二、线下作业

习题1应用题1， 2， 5， 7