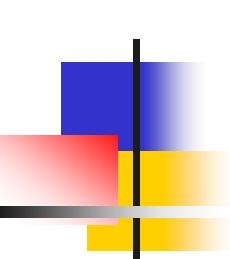


# Chap1 操作系统引论

## ■ 内容

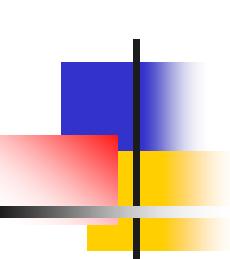
- 操作系统的目标和作用
- 操作系统的发展过程
- 操作系统的基本特性
- 操作系统的主要功能
- OS结构设计
- 常见的OS



# Chap1 操作系统引论

## ■ 目的及要求

- 了解操作系统的目  
标、作用和模型
- 领会和理解操作系  
统的发展过程
- 初步了解和领会操  
作系统的特征和服务
- 领会和掌握操作系  
统的功能
- 了解操作系统的进  
一步发展



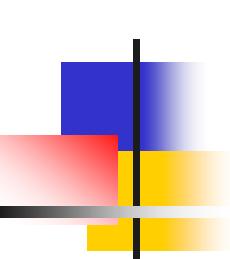
# Chap1 操作系统引论

## ■ 重点

- 批处理操作系统
- 分时操作系统
- 操作系统的特征
- 操作系统的功能

## ■ 难点

- 多道程序设计的基本概念
- 多道批处理系统的特征



# Chap1 操作系统引论

## ■ 内容

- § 1.1 操作系统的目标和作用
- § 1.2 操作系统的发展过程
- § 1.3 操作系统的基本特性
- § 1.4 操作系统的主要功能
- § 1.5 OS结构设计
- § 1.6 常见的OS

# § 1.1 操作系统的目标和作用

## ■ OS的定义

- 你用过哪些OS?
- OS能做什么?

Windows; Unix; Linux; Dos

- 各种命令: dir copy del format
  - 启动、结束用户程序
  - 系统调用: 例如INT指令
  - UNIX 等提供多任务、多用户环境
- 结论:

os为你完成所有 “硬件相关、应用无关” 的工作, 以给你方便、效率、安全

- OS不能做什么?

- 不做天气预报
- 不做房屋设计
- 不是编译程序 ... ...

总之, OS不直接解决最终具体应用问题, 也不负责编译源程序...

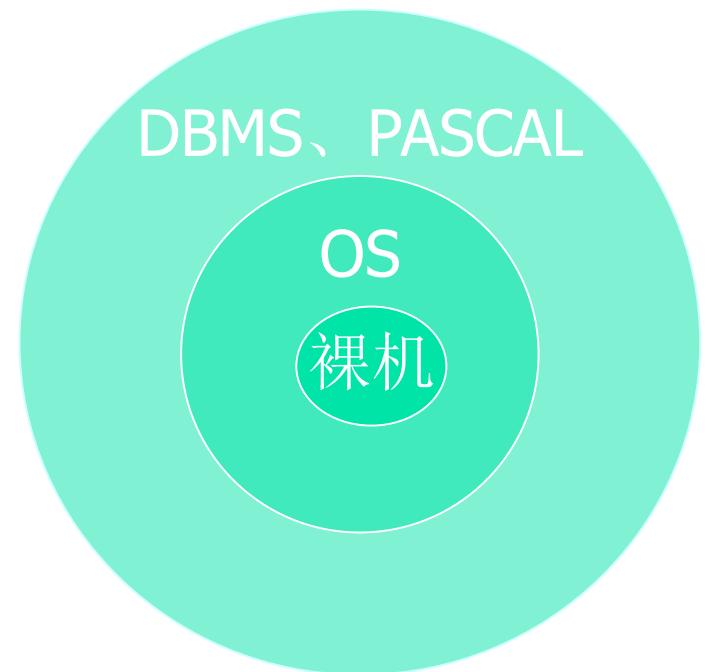
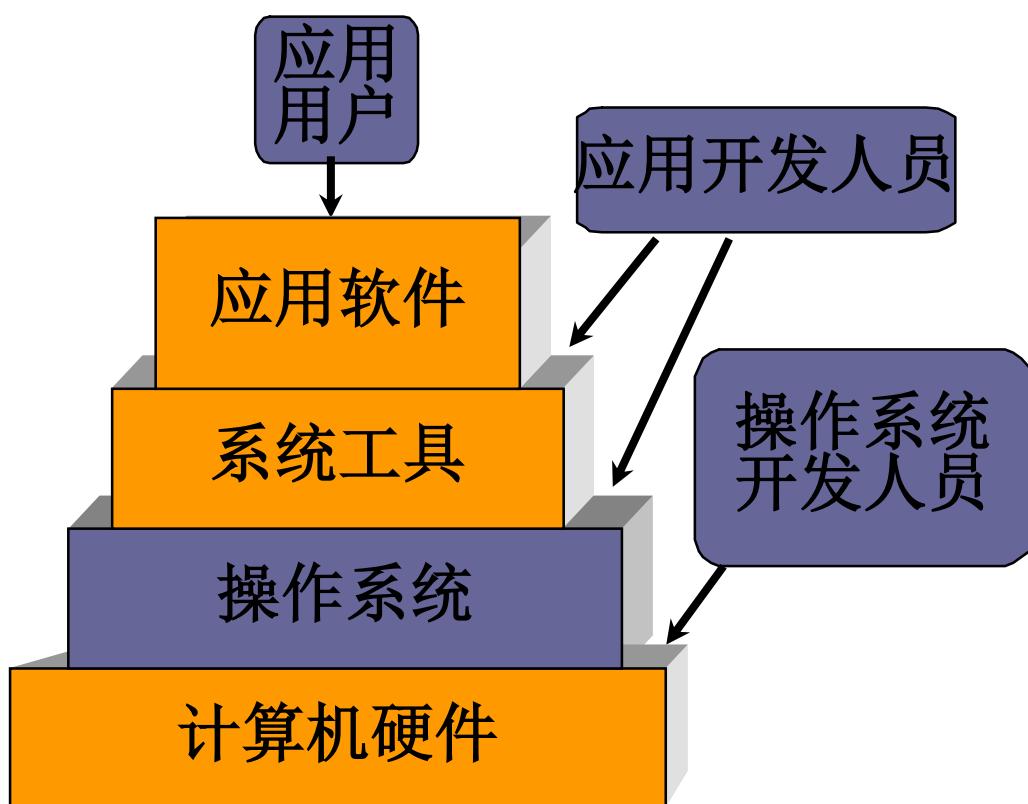
- OS是什么?

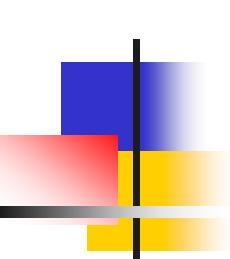
OS是直接控制和管理计算机硬件、软件资源, 合理地对各类作业进行调度, 以方便用户使用的程序集合

# § 1.1 操作系统的目标和作用

## ■ OS在计算机中的地位

紧贴系统硬件之上，所有其他软件之下（是其他软件的共同环境）





# § 1.1 操作系统的目标和作用

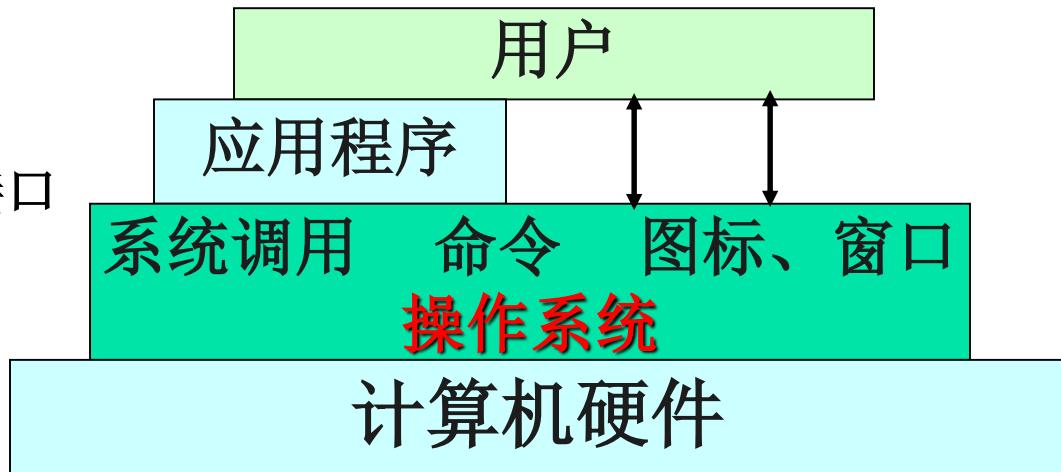
## ■ OS的目标

- 方便性
- 有效性
- 可扩充性
- 开放性

# § 1.1 操作系统的目标和作用

## ■ OS的作用

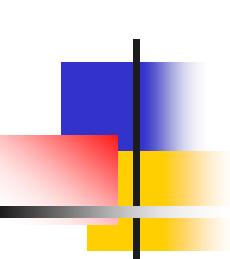
- 作为用户和计算机间的接口



- 作为计算机系统资源的管理者

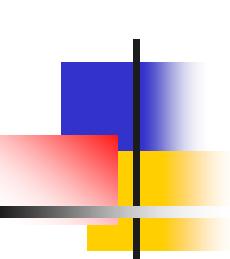


- 用作扩充机器



# § 1.1 操作系统的目标和作用

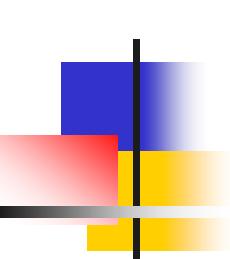
- 推动OS发展的主要动力



# Chap1 操作系统引论

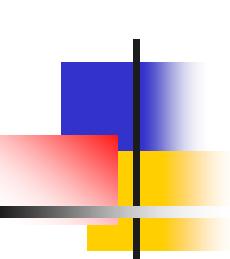
## ■ 内容

- § 1.1 操作系统的目标和作用
- § 1.2 操作系统的发展过程
- § 1.3 操作系统的基本特性
- § 1.4 操作系统的主要功能
- § 1.5 OS结构设计
- § 1.6 常见的OS



## § 1.2 OS的发展过程

- 无OS
- OS分类



## § 1.2 OS的发展过程

- 无OS
- OS分类

## § 1.2 OS的发展过程

- 无OS

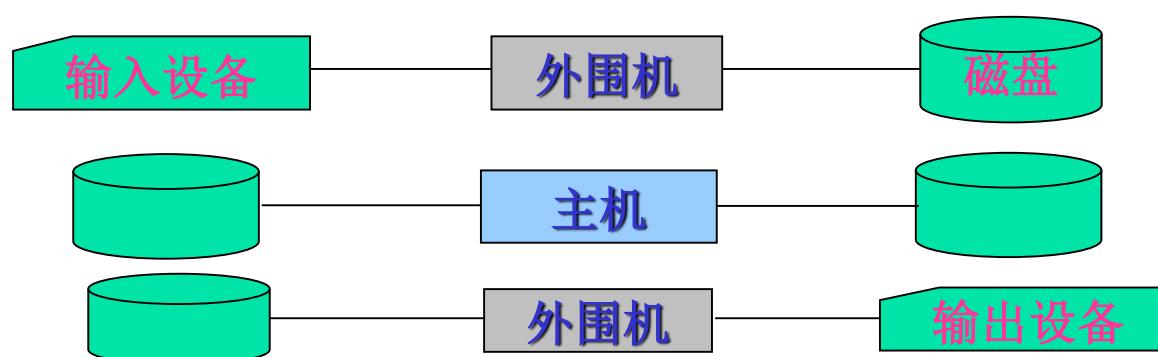
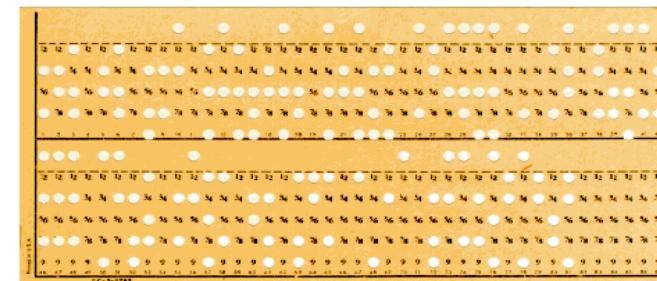
- 人工操作方式

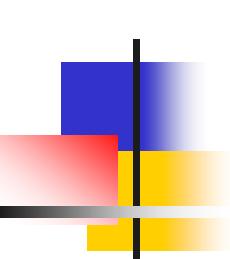
- ⑩ 1946-50年代中：电子管时代，计算机速度慢，无操作系统，集中计算，计算机资源昂贵；

- ⑩ 工作方式：

- ✓ 用户：既是程序员又是操作员；用户是计算机专业人员；
    - ✓ 编程语言：机器语言；
    - ✓ 输入输出：纸带或卡片；

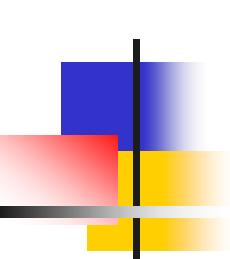
- 脱机I/O方式





## § 1.2 OS的发展过程

- 无OS
- OS分类

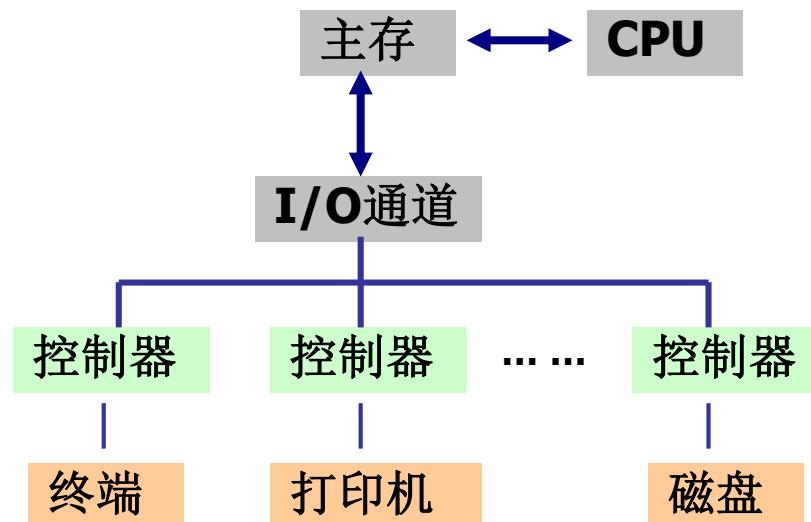


# OS分类

- 批处理系统 (Batch Processing System)
  - 单道
  - 多道
- 分时系统 (Time-Sharing System)
- 实时系统 (Real-Time System)

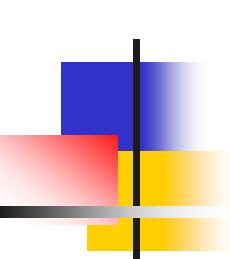
# OS课程所站的角度

- 多任务
- 单CPU
- CPU具有与外设并行操作的能力



特别指出：

1. 程序均由CPU执行
2. CPU只能执行机器语言程序
3. CPU执行的程序均在内存中

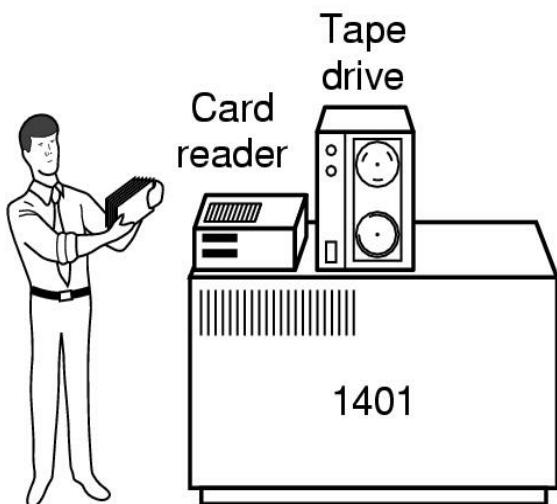


# OS分类

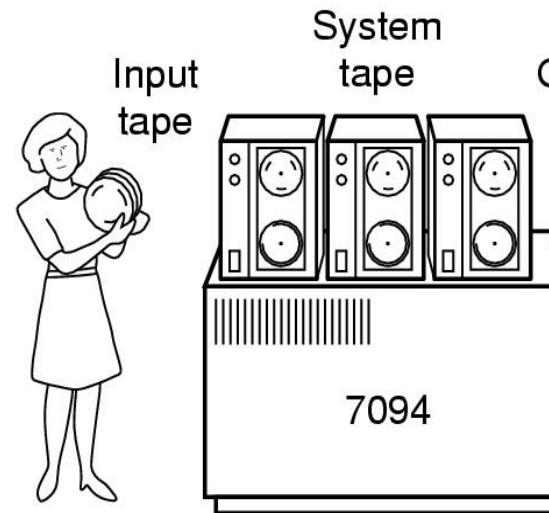
- 批处理系统 (**Batch Processing System**)
- 分时系统 (**Time-Sharing System**)
- 实时系统 (**Real-Time System**)

# 批处理系统

## ■ 批处理系统



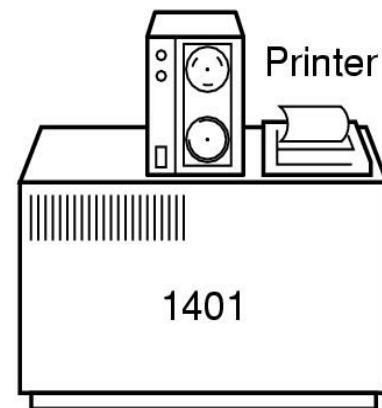
(a)



(c)



(e)



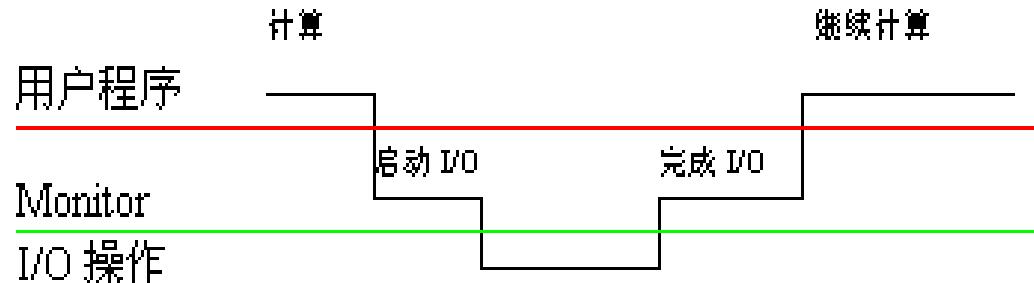
(f)

单道批处理系统

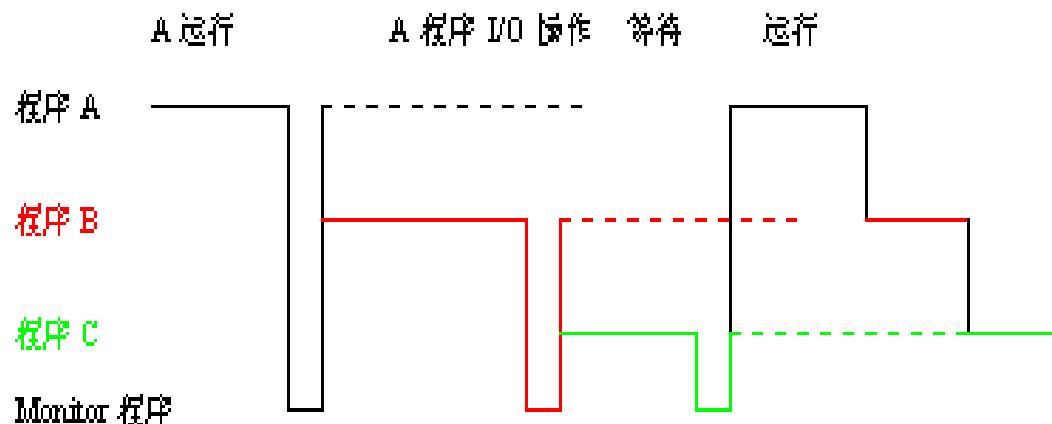
多道批处理系统

# 批处理系统

单道程序工作示例：



多道程序工作示例：

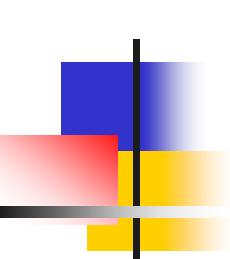


# 批处理系统举例

- 举例：假设一个计算机系统有**256k**主存供用户使用，一个磁盘、一个终端、一台打印机。三个作业：**JOB<sub>1</sub>**、**JOB<sub>2</sub>**、**JOB<sub>3</sub>**。各作业运行时间为**5分钟**、**15分钟**和**10分钟**。它们对资源的具体使用情况如下所示。

作业编号	<b>JOB<sub>1</sub></b>	<b>JOB<sub>2</sub></b>	<b>JOB<sub>3</sub></b>
作业类型	计算型	I/O型	I/O型
占用主存	<b>50k</b>	<b>100k</b>	<b>80k</b>
需磁盘情况	<b>NO</b>	<b>NO</b>	<b>Yes</b>
需终端情况	<b>NO</b>	<b>Yes</b>	<b>NO</b>
需打印机情况	<b>NO</b>	<b>NO</b>	<b>Yes</b>
运行所需时间	<b>5分钟</b>	<b>15分钟</b>	<b>10分钟</b>

- 试比较单道、多道批处理系统时作业的执行时间



# 批处理系统举例

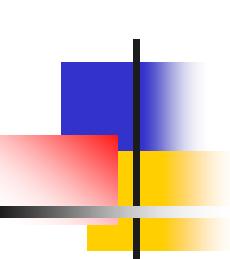
- 由上看出， $JOB_2$ 主要使用终端（键盘和显示器）进行I/O， $JOB_3$ 主要使用磁盘和打印机。 $JOB_2$ 和 $JOB_3$ 需要较少的CPU时间。
- 对于简单批处理，这些作业是按顺序执行。30分钟后三个作业全部完成。（ $JOB_1$ 运行5分钟完成， $JOB_2$ 等待5分钟再用15分钟完成，20分钟后， $JOB_3$ 开始执行）
- 若采用多道程序设计技术，让三个作业同时装入主存并运行。由于它们运行中几乎不同时使用同类资源，在 $JOB_1$ 进行计算的同时 $JOB_2$ 可在终端上进入输入/输出，而 $JOB_3$ 在使用磁盘和打印机。

# 批处理系统举例

- 多道程序与单道程序的对比

	单道	多道(三道作业)
完成所需时间	<b>30分钟</b>	<b>15分钟</b>
处理机利用率	<b><math>5/(5+15+10)=17\%</math></b>	<b><math>5/15=33\%</math></b>
存贮器利用率	<b><math>(50/256+100/256+80/256)/3=30\%</math></b>	<b><math>(50+100+80) / 256=90\%</math></b>
磁盘利用率	<b><math>10/30=33\%</math></b>	<b><math>10/15=67\%</math></b>
打印机利用率	<b><math>10/30=33\%</math></b>	<b><math>10/15=67\%</math></b>
终端利用率	<b><math>15/30=50\%</math></b>	<b><math>15/15=100\%</math></b>
吞吐量	<b><math>3/0.5=6 \text{ jobs/小时}</math></b>	<b><math>3/0.25=12 \text{ jobs/小时}</math></b>
平均周转时间	<b><math>(5+20+30)/3=18 \text{分钟}</math></b>	<b><math>(5+15+10)/3=10 \text{分钟}</math></b>

- 多道程序运行，系统性能的改善是明显的。将多道程序设计技术应用于批处理系统，就形成多道批处理系统。



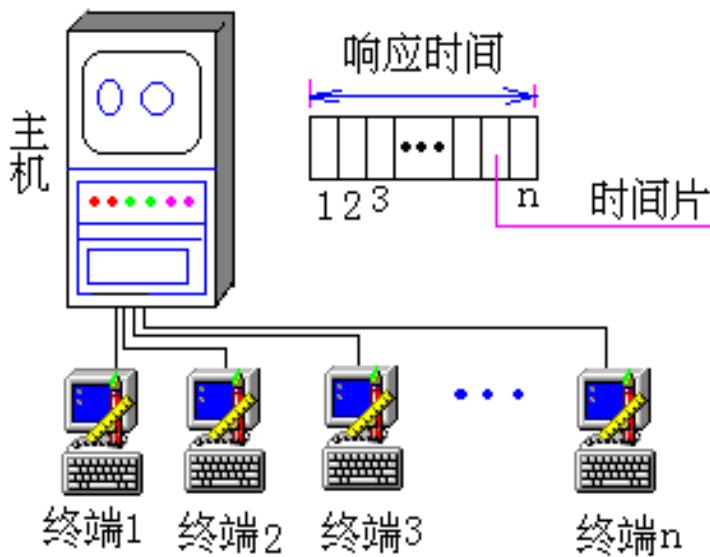
# OS分类

- 批处理系统 (Batch Processing System)
- 分时系统 (Time-Sharing System)
- 实时系统 (Real-Time System)

# 分时系统(Time-Sharing System)

## 主要动力：用户的需求

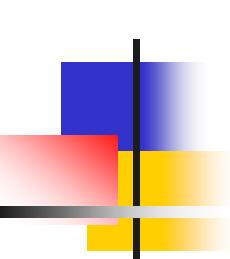
- 目标：对用户的请求及时响应；尽量提高系统资源的利用率
- 原理：



- 基本特征：多路性；独立性；及时性；交互性；

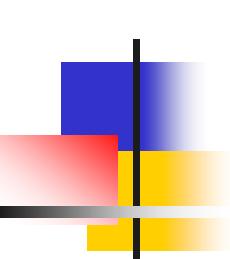
➤一台计算机连接多个终端，用户通过各自的终端把作业送入计算机；计算机又通过终端向各个用户报告其作业的运行情况。

➤计算机能**分时轮流**地为各终端用户提供服务，并能及时地对用户服务请求予以响应。



# OS分类

- 批处理系统 (Batch Processing System)
- 分时系统 (Time-Sharing System)
- 实时系统 (Real-Time System)



# 实时系统(Real-Time System)

- 目标：提高系统的响应时间，对随机发生的外部事件作出及时响应并对其进行处理。

- 分类：

实时控制系统

要求计算机能尽快处理测量系统测得的数据，以尽快实施响应控制。如：工业控制；导弹发射；飞机飞行

实时信息系统

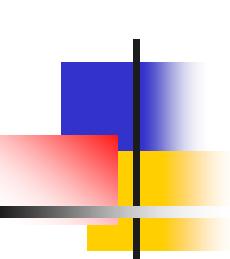
要求计算机能对终端设备发来的服务请求及时予以正确的回答。如：订票系统；情报检索系统

- 基本特征： 快速的响应时间； 有限的交互能力；  
高可靠性；

# OS分类

三种基本操作系统的比较：

	多路性	独立性	及时性	交互性	可靠性
批处理系统	无	无	差 (天,时)	差	一般
分时系统	多终端服务	有	好 (分,秒)	好	可靠
实时系统	多路采集、多路控制	有	最好 (ms,μs)	一般	高度可靠



# Chap1 操作系统引论

## ■ 内容

- § 1.1 操作系统的目标和作用
- § 1.2 操作系统的发展过程
- § 1.3 操作系统的基本特性
- § 1.4 操作系统的主要功能
- § 1.5 OS结构设计
- § 1.6 常见的OS

# § 1.3 操作系统的基本特性

## ■ 并发(Concurrency):

并发: 指两个或多个事件在同一时间间隔内发生。

并行: 指两个或多个事件在同一时刻发生。

## ■ 共享(Sharing):

指系统中的资源供内存中的多道程序所共同使用。

分类 {  
    互斥共享方式: 例 打印机  
    同时访问方式: 例 磁盘

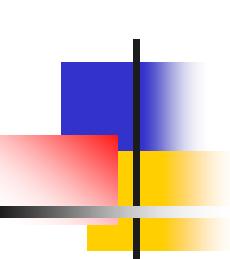
## ■ 虚拟(Virtual):

指通过某种技术把一个物理实体变成若干个逻辑上的对应物。

例, 分时系统将1机虚拟为多机 ←———— 由于共享资源

## ■ 异步性(Asynchronism):

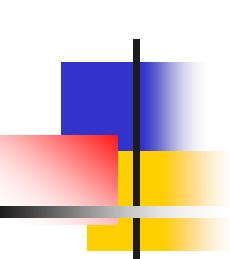
系统中并发执行的多道程序“走走停停”, 以不可预知的速度向前推进



# Chap1 操作系统引论

## ■ 内容

- § 1.1 操作系统的目标和作用
- § 1.2 操作系统的发展过程
- § 1.3 操作系统的基本特性
- § 1.4 操作系统的主要功能
- § 1.5 OS结构设计
- § 1.6 常见的OS



## § 1.4 操作系统的主要功能

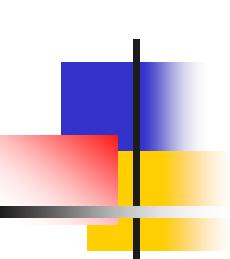
OS定义：

OS是直接控制和管理计算机硬件、软件资源，合理地对各类作业进行调度，以方便用户使用的程序集合

- 处理机管理功能
- 存储器管理功能
- 设备管理功能
- 文件管理功能
- 用户接口

OS的主要任务：

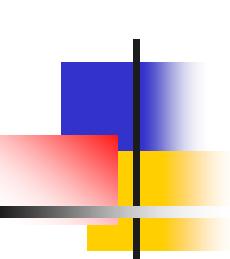
为多道程序的运行提供良好的运行环境，以保证多道程序能有条不紊、高效地运行，并能最大程度地提高系统中各种资源的利用率和方便用户的使用。



# Chap1 操作系统引论

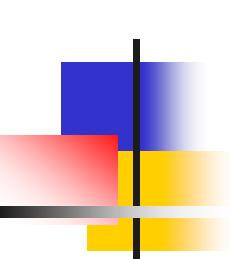
## ■ 内容

- § 1.1 操作系统的目标和作用
- § 1.2 操作系统的发展过程
- § 1.3 操作系统的基本特性
- § 1.4 操作系统的主要功能
- § 1.5 OS结构设计
- § 1.6 常见的OS



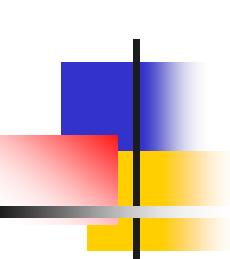
## § 1.5 OS结构设计

- 操作系统的结构设计经历了以下几代：
  - 传统的操作系统结构
    - 无结构操作系统
    - 模块化OS结构
    - 分层式OS结构
  - 现代操作系统结构
    - 微内核的OS结构



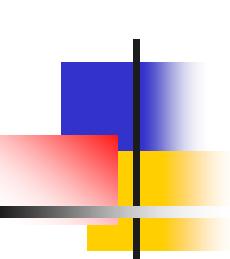
# 无结构操作系统

- OS是由众多的过程直接构成，各过程之间可相互调用，但OS内部不存在任何结构，所以这种OS是无结构的，又称为整体系统结构。
- 缺点：  
既庞大又杂乱，缺乏清晰的程序结构；程序错误多，调试难、阅读难、理解难、维护难。



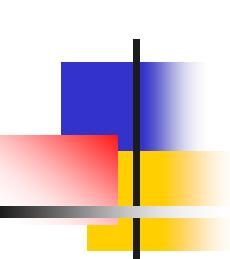
## § 1.5 OS结构设计

- 操作系统的结构设计经历了以下几代：
  - 传统的操作系统结构
    - 无结构操作系统
    - 模块化OS结构
    - 分层式OS结构
  - 现代操作系统结构
    - 微内核的OS结构



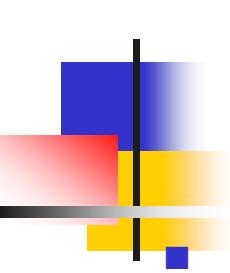
# 模块化操作系统结构

- OS是采用“模块化程序设计”技术，按其功能划分为若干个独立的模块，管理相应的功能，同时规定好各模块之间的接口，以实现其交互，对较大模块又可按子功能进一步细分下去。
- 优点：提高了OS设计的正确性、可理解性和可维护性  
增强了OS的可适用性
- 缺点：加速了OS的开发过程  
模块及接口划分较困难  
从功能上划分模块，未区别共享资源和独占资源  
由于管理的差异，使OS结构变得不够清晰  
例，Linux（多用户、多任务OS）



## § 1.5 OS结构设计

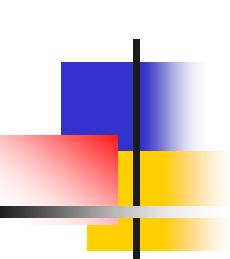
- 操作系统的结构设计经历了以下几代：
  - 传统的操作系统结构
    - 无结构操作系统
    - 模块化OS结构
    - 分层式OS结构
  - 现代操作系统结构
    - 微内核的OS结构



# 分层式操作系统结构

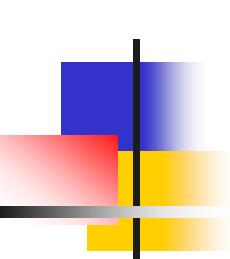
- 分层式OS结构是对模块化结构的一种改进，它按分层式结构设计的基本原则，将OS划分为若干个层次，每一层都只能使用其底层所提供的功能和服务，从硬件开始，在其上面一层一层地自底向上增添相应功能的软件，这种OS结构称为分层式OS结构。

- 优点：
  - 每一步设计都建立在可靠的基础上，结构更清晰
  - 调试和验证更容易，正确性更高
- 缺点：
  - 系统效率降低。只能是单向依赖，每执行一个功能，自底向上需要穿越很多个层次。



## § 1.5 OS结构设计

- 操作系统的结构设计经历了以下几代：
  - 传统的操作系统结构
    - 无结构操作系统
    - 模块化OS结构
    - 分层式OS结构
  - 现代操作系统结构
    - 微内核的OS结构



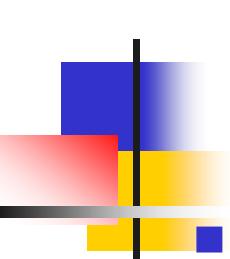
# 微内核的OS结构

- 微内核技术的主要思想

在OS内核中只留下一些最基本的功能，而将其他服务分离出去，由工作在用户态下的进程来实现，形成所谓“客户/服务器”模式。客户进程可通过内核向服务器进程发送请求，以取OS的服务。

- 微内核

精心设计的，能实现现代OS核心功能的小型内核，它小而精炼，运行在核心态下，开机后常驻内存，不会因内存紧张而换出，它为构建通用OS提供了一个重要基础。



# 微内核的OS结构

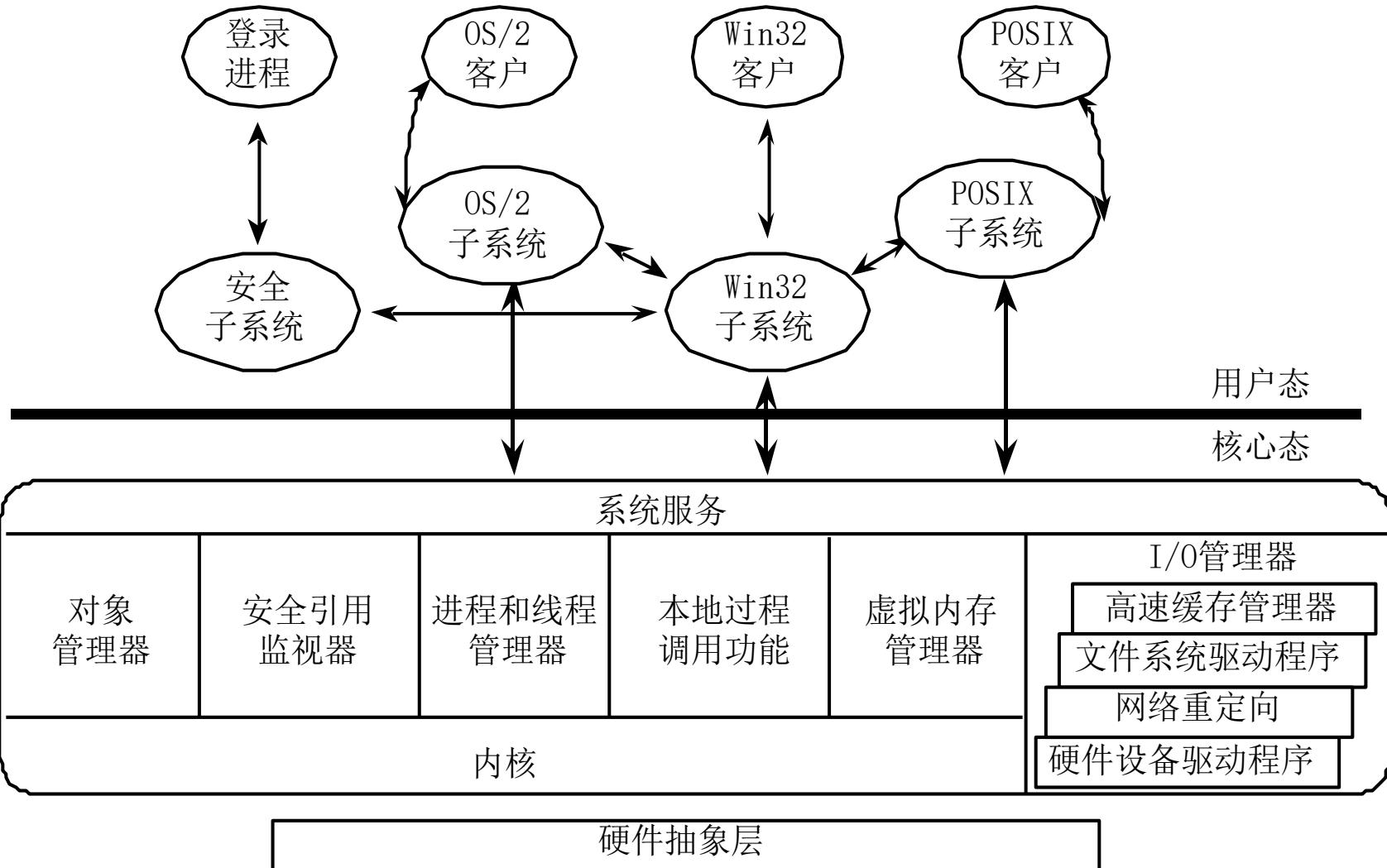
## 微内核的基本功能

- 进程管理
- 存储器管理
- 进程通信管理
- I/O设备管理

## ■ 特点

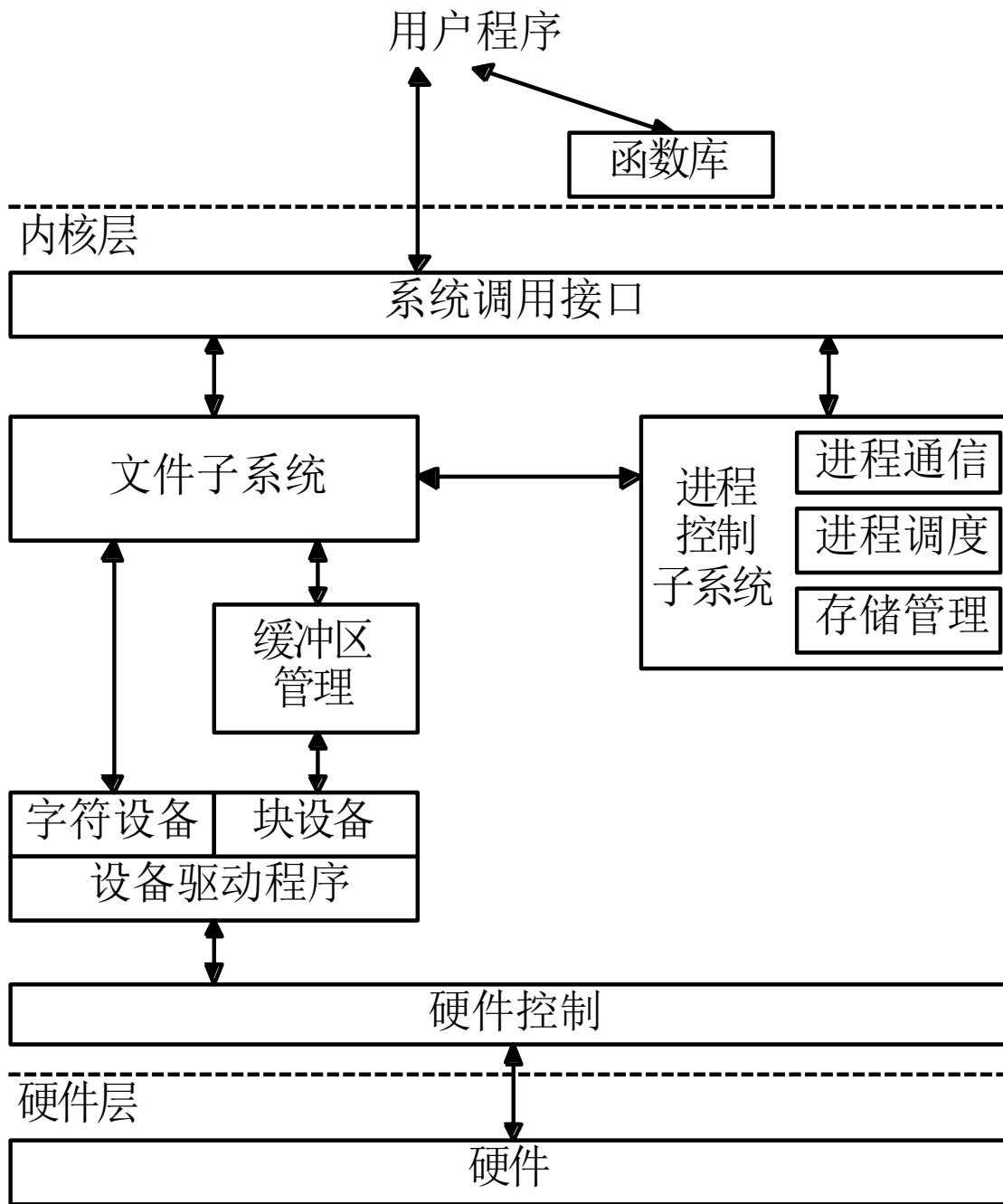
- 小而精练
- 系统的灵活性和可扩充性好
- 系统的可靠性高
- 适用于分布式系统

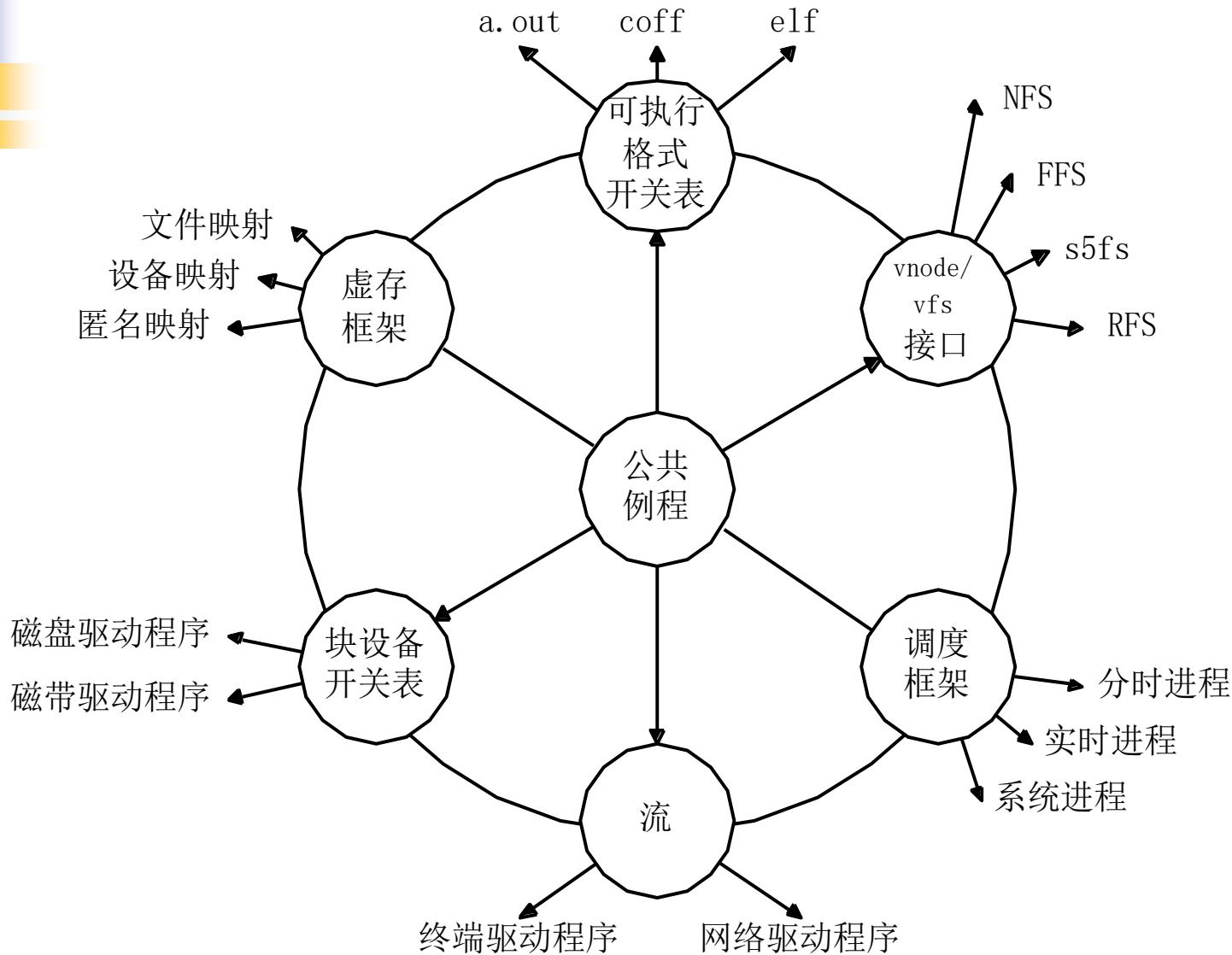
例，windows 2000/XP、UNIX、嵌入式OS



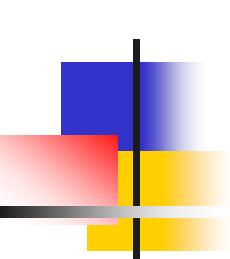
## Windows NT 体系结构

# 传统的UNIX结构





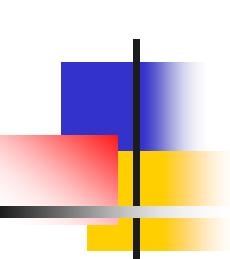
## 现代UNIX结构



# Chap1 操作系统引论

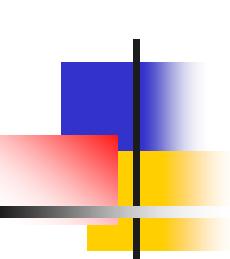
## ■ 内容

- § 1.1 操作系统的目标和作用
- § 1.2 操作系统的发展过程
- § 1.3 操作系统的基本特性
- § 1.4 操作系统的主要功能
- § 1.5 OS结构设计
- § 1.6 常见的OS



## § 1.6 常见的OS

- MS DOS
- MS Windows
- UNIX
- Linux
- 手持系统 (handheld system)
- 嵌入式操作系统 (Embedded OS)



# MS DOS

**IBM PC, CPU 8088/8086, BIOS**, 单用户单任务, 简单分层结构, **16位**

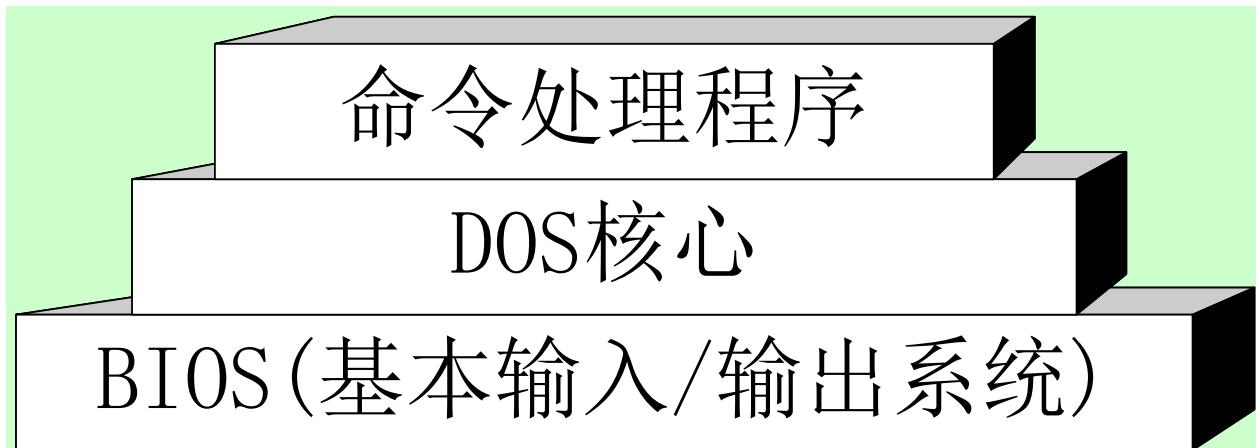
## ■ MS DOS的历史

- DOS用过的名字包括**QDOS**、**86-DOS**、**MS-DOS**和**PC-DOS**;
- **1981年8月**: **PC-DOS 1.0**: 第一个**DOS**版本;
- **1982年6月**: **PC-DOS 1.1**: **IBM PC**, 支持软盘的个人操作系统;
- **1983年3月**: **DOS2.0**: **PC XT**, 支持硬盘和目录的层次结构;
- **1984年8月**: **DOS3.0**: **PC AT (Intel 80286 CPU)**;
- **1987年4月**: **DOS3.3**: 提供对**IBM PS/2**的支持 (如**3.5"软驱**) ;
- **1988年8月**: **DOS4.0**: 支持大于**32M**的硬盘;
- **1991年6月**: **DOS5.0**: 改进对扩展内存的支持;
- **1993年3月**: **MS-DOS6.0**: 支持磁盘压缩;
- **1994年5月**: **MS-DOS6.22**: 提供新的磁盘压缩方法;
- **1995年8月**: **MS-DOS7.0**: 作为**Windows95**的一部分;

# MS DOS

## MS DOS的结构

- DOS BIOS(Basic Input/Output System): 由一组与硬件相关的设备驱动程序组成，实现基本的I/O功能；
- DOS核心：提供一套独立于硬件的系统功能：内存管理、文件管理、字符设备和输入/输出、实时时钟等；
- 命令处理程序：对用户命令进行分析和执行；



# MS Windows

**CPU: 80286, 80386, 80486, Pentium, ...**

单用户或多用户多任务（分时系统），**16位/16和32位混合/32位**

## ■ Windows的历史

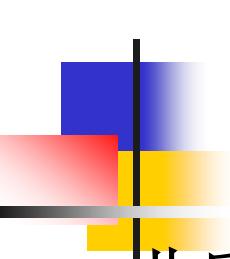
- **1983年11月：** Microsoft宣布**Windows**的第一个版本；以字符为基础的窗口系统；
- **1985年11月：** **Windows 1.0**；
- **1990年5月：** **Windows 3.0**（成功版本），**16位OS**，借鉴**Apple Macintosh**给出友好的用户界面；
- **1992年10月：** **Windows For WorkGroups 3.1**；
- **1993年11月：** **Windows For WorkGroups 3.11**；
- **1995年8月：** **Windows 95**：内置浏览器；
- **1998年7月：** **Windows 98**；
- **2000年9月：** **Windows ME**；
- **1993年：** **Windows NT 3.1**, **32位OS**，支持**DOS**和**Windows**应用程序；
- **1994年4月：** **Windows NT 3.51**；
- **1996年8月：** **Windows NT 4.0**；
- **1999年12月：** **Windows 2000 (Professional, Server, Advanced Server)**, **32位OS**；
- **2001年10月：** **Windows XP**
- **2007年1月：** **Windows Vista**

# UNIX

多用户多任务，**16/32/64位**,**BSD, SVR4**（模块式结构），**OSF/1**（微内核结构）

## ■ 诞生

- **1965年**，麻省理工学院（**MIT**）、通用电子公司（**GE**）及**AT&T**的贝尔实验室联和开发了一个叫做**Multics**（多路复用信息与计算服务）的实验操作系统。
- **1969年**，**AT&T**贝尔实验室的肯.汤姆森（**Ken Thompson**）和丹尼斯.里奇（**Dennis Ritchie**）和其他的研究人员在**DEC PDP-7**上开发出最初的**UNIX**系统雏形—**UNICS**（单道信息与计算服务），后来谐音拼写成**UNIX**。
- **1970年**，**UNIX**被移植到**PDP-11**上，第一次得到实用。此时的系统全部用汇编语言写成。
- **1971年**，丹尼斯.里奇发明了C语言。
- **1973年**，肯.汤姆森和丹尼斯.里奇用C语言对**UNIX**核心进行重写，使系统便于理解、修改和移植。



# UNIX

此后**UNIX**逐渐演变为两大分支：

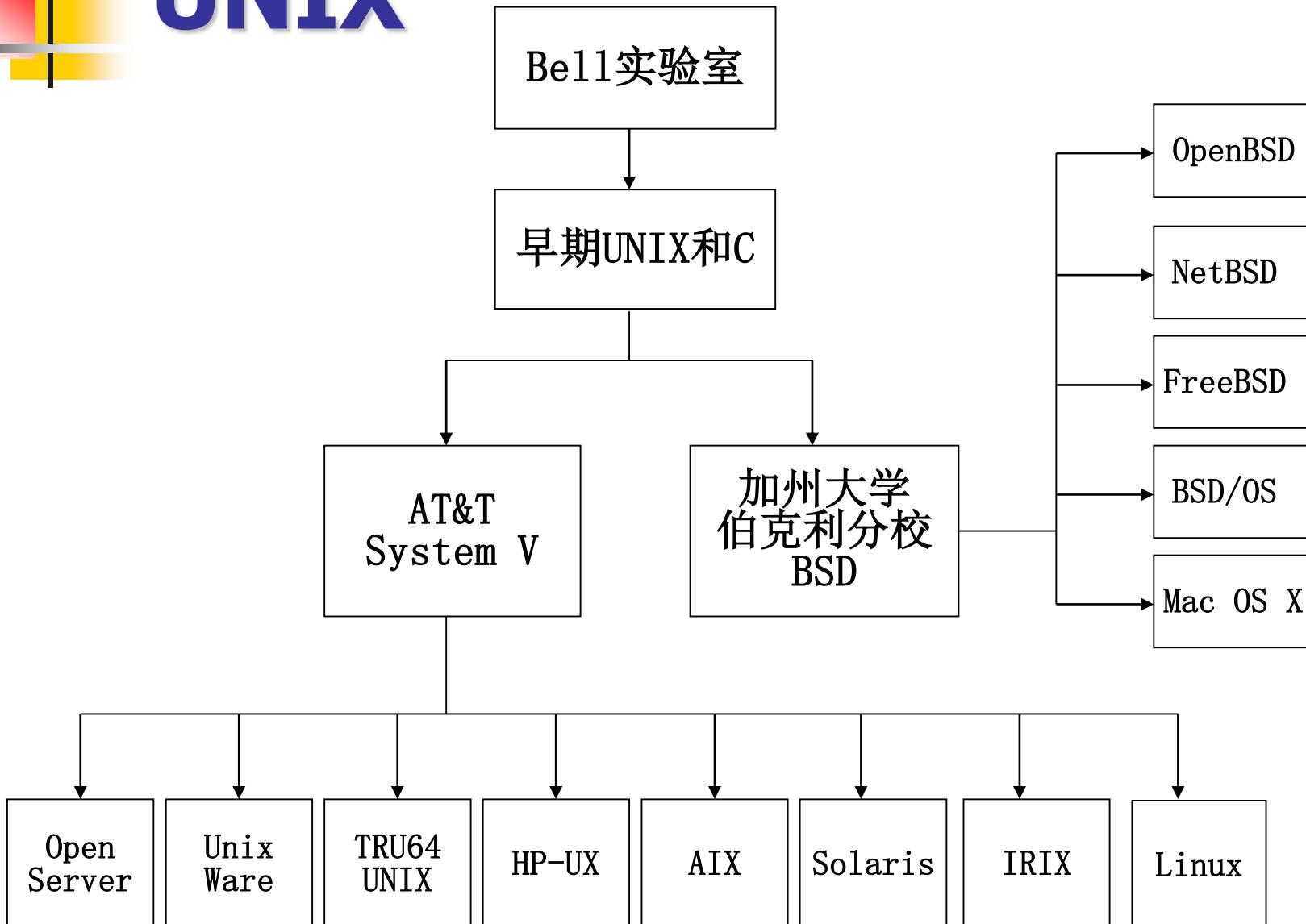
■ AT&T的**System V**

- 1983年, AT&T推出**UNIX System V Release 1**
- 1984年, AT&T推出**UNIX System V Release 2**
- 1987年, AT&T推出**UNIX System V Release 3**
- 1990年, AT&T推出**UNIX System V Release 4**
- 1992年, AT&T推出**UNIX SVR 4.2**

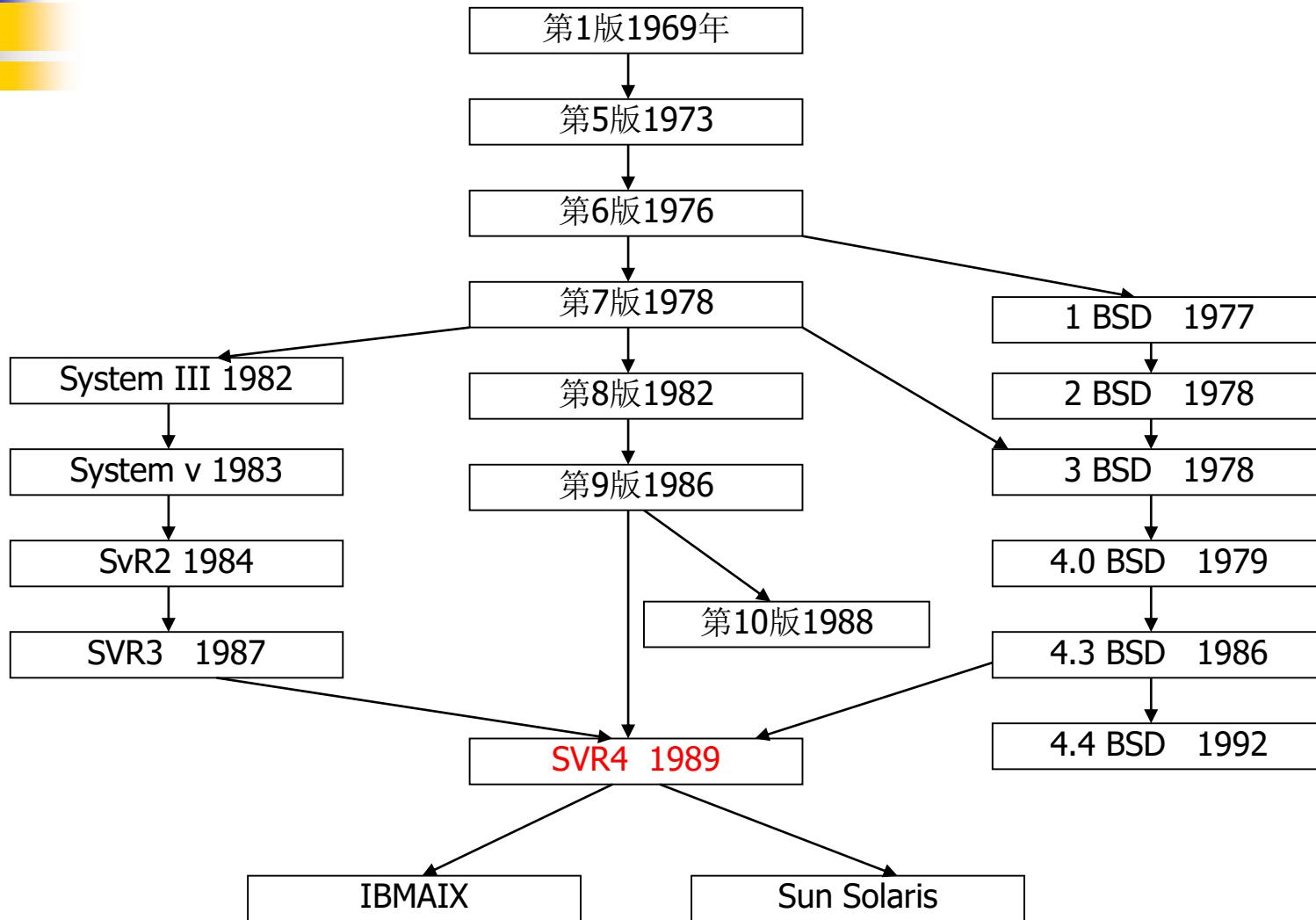
■ 加州大学伯克利分校的**BSD** (**Berkeley Software Distribution**)

- 1975年, Berkeley发布 **1BSD**
- 1978年, Berkeley发布 **2BSD**
- 1979年, Berkeley发布 **3BSD**
- 1980年, Berkeley发布 **4BSD**
- 1983年, Berkeley发布 **4.1BSD**
- 1984年, Berkeley发布 **4.2BSD**
- 1987年, Berkeley发布 **4.3BSD**
- 1993年, Berkeley发布 **4.4BSD**

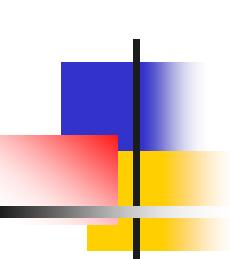
# UNIX



# Bell Labs



**UNIX版本发展历史**



# 开放软件

## ■ 通用开放软件环境（COSE）

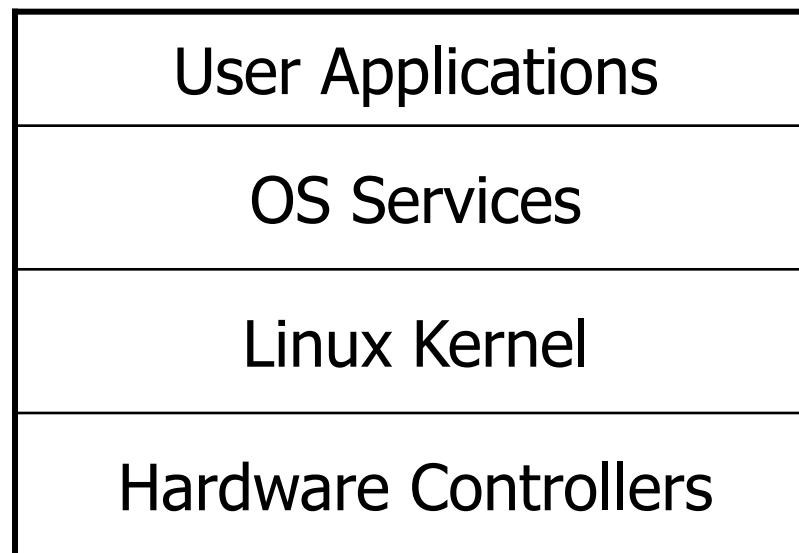
一些主要的**UNIX**系统厂商在1993年成立了通用开放软件环境联盟。这些厂家当时包括**HP**, **IBM**, **Sun**, **SCO**, **Novell**及**UNIX**系统实验室。

## ■ 自由软件基金会（FSF）

- 自由软件基金会（Free Software Foundation, FSF）的目标是创建一个与**UNIX**类似的系统，名为**HURD**。
- 他们的这项工程称为**GNU**（全称为**GNU's Not UNIX**）。
- 在编写**HURD**的过程中，他们编写了许多非常有用的、可移植性很强的流行实用程序。
- 例如**GNU C**编译器**gcc**。用户既可以得到这些工具的源代码而自行进行编译，也可以得到能立即运行的可执行文件。

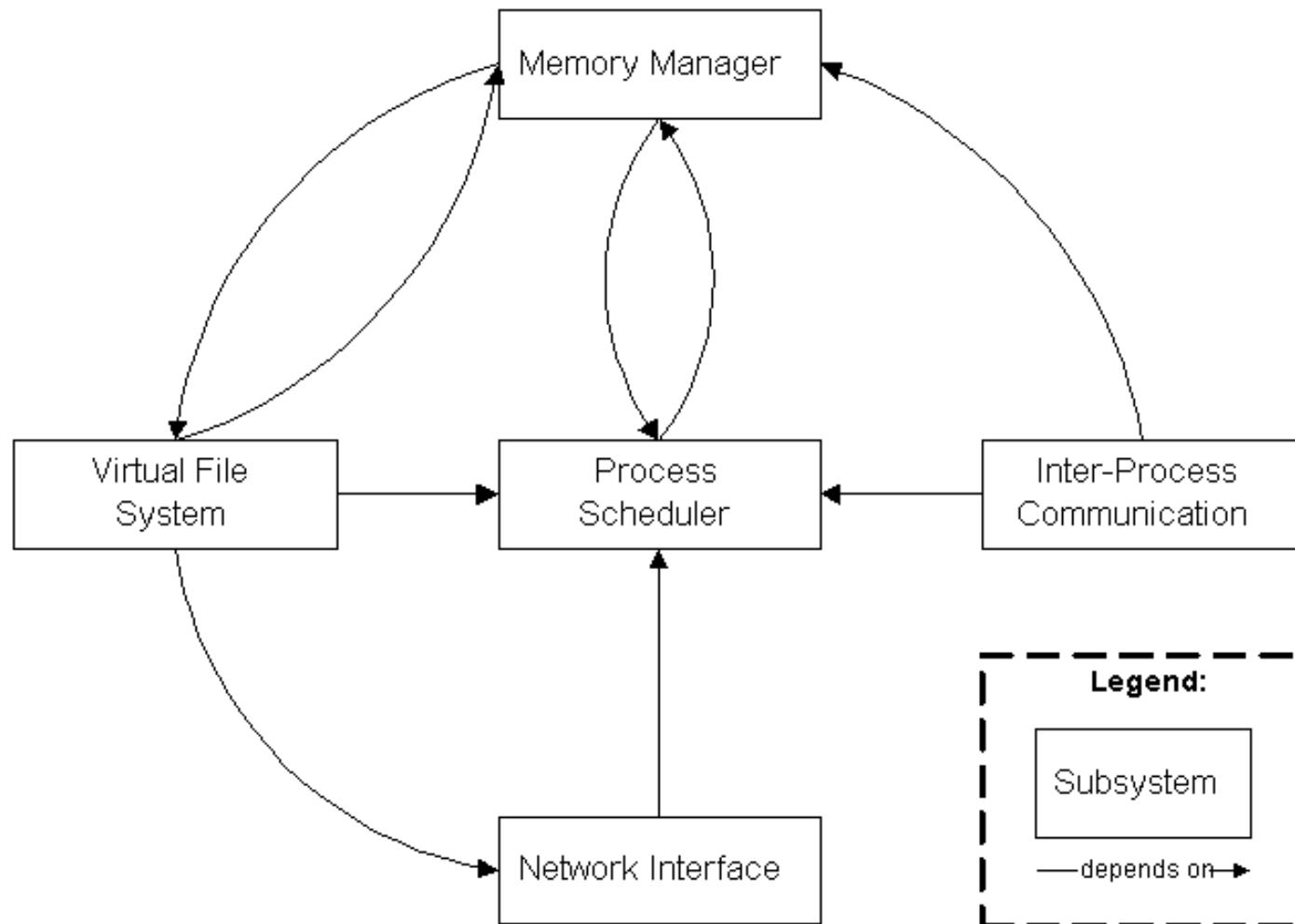
# Linux

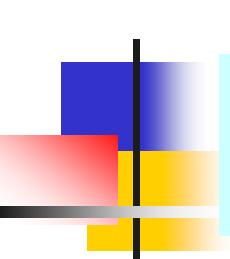
- Linux的体系结构
  - 用户应用：文字处理工具、浏览器等用户应用程序；
  - 系统服务：它是操作系统的一部分，但不是内核。包括窗口系统、命令解释程序、应用开发工具（编译器、系统库等）
  - Linux内核：操作系统内核，完成资源管理功能。
  - 硬件控制：系统中的所有物理设备控制。



# Linux

## Linux的内核结构

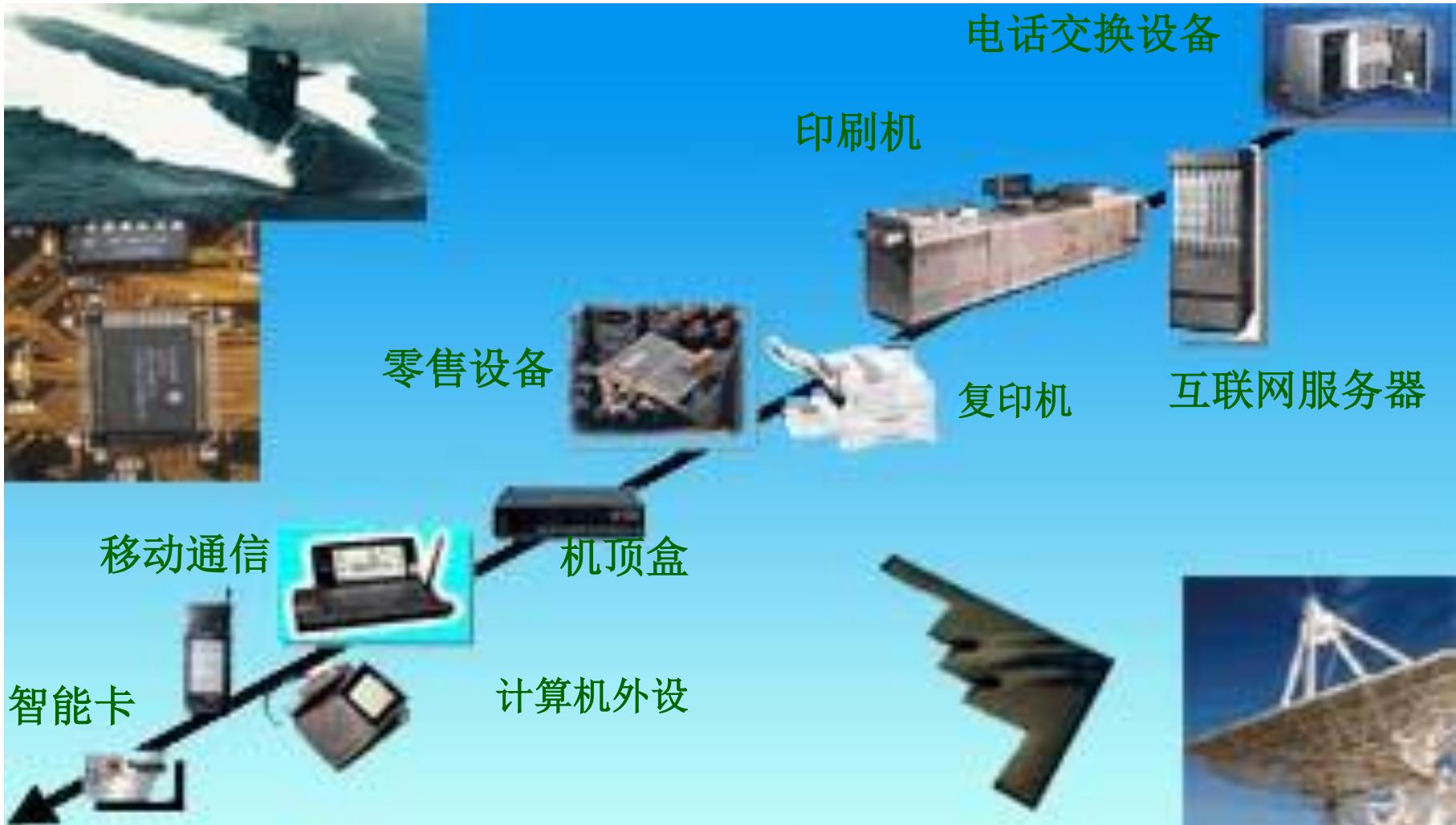


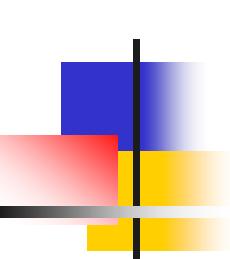


# 手持系统 (handheld system) 嵌入式操作系统 (Embedded OS)

- 个人数字助理 (Personal Digital Assistants, PDAs)
- 蜂窝电话 (Cellular telephones)
- 存在的问题
  - 内存有限(32M – 64M)
  - 低速处理器 (只有个人计算机处理器速度的几分之一)
  - 屏幕小 (5英寸×3英寸)

# 嵌入式操作系统 (Embedded OS) 的应用





# 作业