

# 第1章 操作系统引论 (5学时)



主讲教师: 张春元

联系电话: 13876004640

课程邮箱: [haidaos@126.com](mailto:haidaos@126.com)

邮箱密码: zhangchunyuan



# 本章主要内容

- ❖ 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 OS (Operating Systems)的主要功能
- ❖ 1.5 OS结构设计





# 1.1 操作系统的目标、作用

- ❖ 1.1.1 操作系统的目标
- ❖ 1.1.2 操作系统的作用
- ❖ 1.1.3 操作系统发展的主要动力



# 1.1 操作系统的目标、作用

## ❖ 操作系统的定义

- \* OS是一组控制和管理计算机软硬件资源、合理地各类作业进行调度以及方便用户使用的程序集合。

## ❖ 操作系统的位置

- \* OS是配置在计算机上的第一层软件，是对硬件的首次扩充。

## ❖ 操作系统的地位

- \* OS在计算机系统中占据特别重要的地位，是计算机中最重要的系统软件，是其它系统软件和应用软件运行的基础。



## 1.1.1 操作系统的目标

### ❖ 有效性

- \* (1) 提高计算机系统资源的利用率;
- \* (2) 提高计算机系统吞吐量。

设计操作系统的  
两大最重要目标

### ❖ 方便性

- \* 提供良好的用户接口（如图形化界面、命令），隐藏硬件系统的差别，易于使用。

### ❖ 可扩充性

- \* 必须便于增加和修改功能和模块，以适应计算机硬件、体系结构以及应用发展的要求。

### ❖ 开放性

- \* 开放性是指系统能遵循世界标准规范，特别是遵循开放系统互联（OSI）国际标准。OS必须提供统一开放的环境，具有可移植性和互操作性。



## 1.1.2 操作系统的作用

### ❖ 作用1：作为用户与计算机硬件系统之间的接口

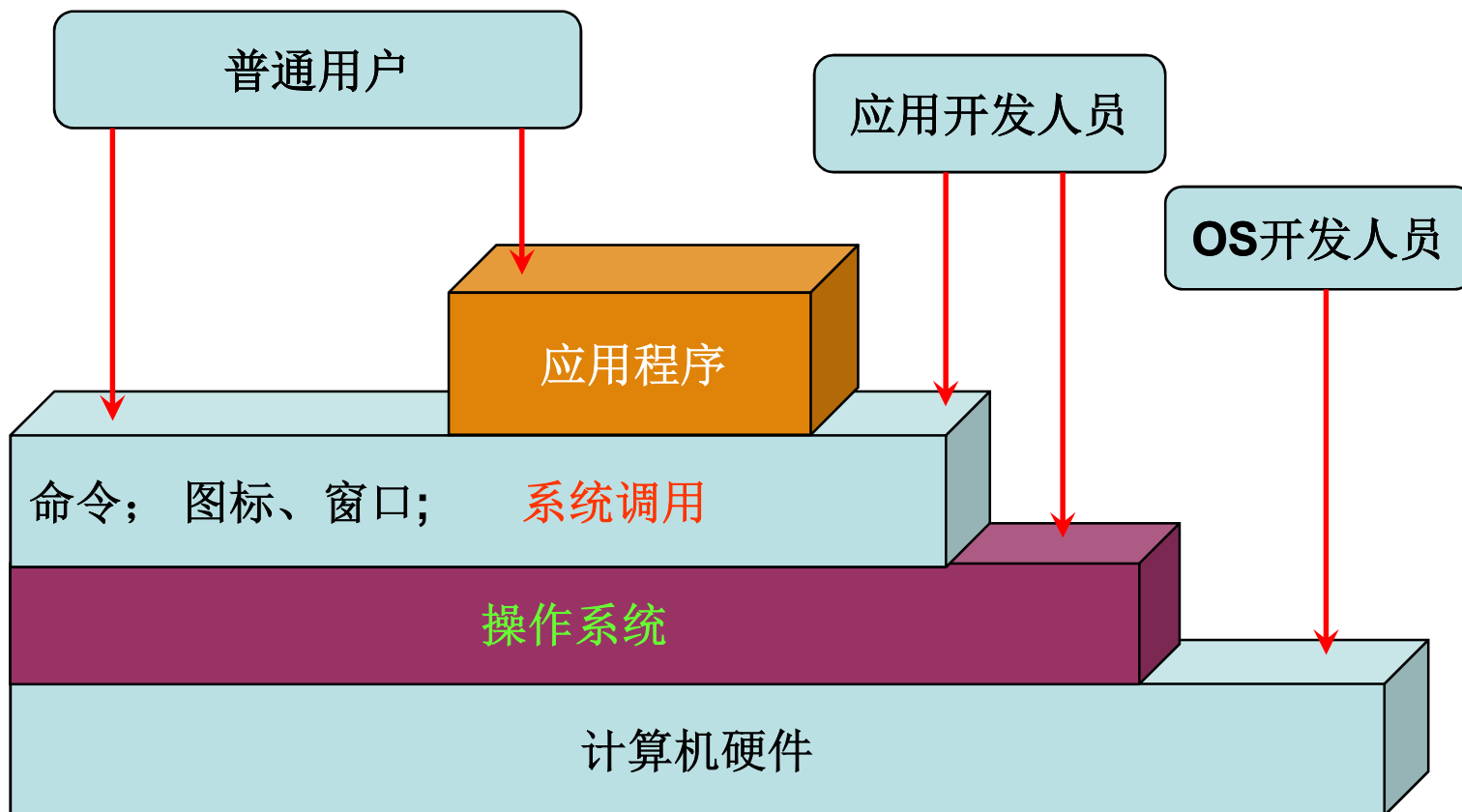


图1—1 OS作为接口的示意图

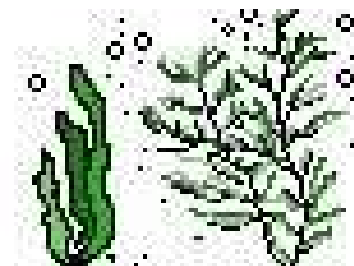


## 1.1.2 操作系统的作用

### ❖ 作用2：作为计算机系统资源的管理者

\* 四类资源：处理机、存储器、I/O(Input/Output)设备、信息（数据和程序）

- ✓ 处理机管理：分配和控制处理机
- ✓ 存储器管理：分配及回收内存
- ✓ I/O设备管理：I/O设备分配与操作
- ✓ 文件管理：文件存取、共享和保护





## 1.1.2 操作系统的作用

### ❖ 作用3: 实现了对计算机资源的抽象

- \* 扩充机器或虚拟机是指计算机为了使用户在不必涉及和了解其硬件工作细节的情况下能方便地使用而向用户提供的一种虚拟的扩展计算机。
- \* 分层扩充，层层抽象，屏蔽细节。







## 1.1.3 操作系统发展的主要动力

- ❖ 追求更高的计算机资源利用率
  - \* 例如多道批处理系统的出现; SPOOLing; 虚存
- ❖ 方便用户
  - \* 如分时交互式系统的出现、图形界面OS
- ❖ 器件的不断更新换代
  - \* CPU: 8位 - 16位 - 32位 - 64位; 多核; 外设
- ❖ 计算机体系结构的不断发展
  - \* 单核OS - 多核OS - 网络OS - ...
- ❖ 不断提出新的应用需求

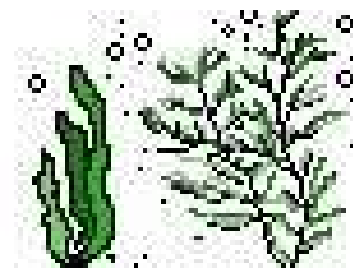




## 1.2 操作系统的发展过程

- ❖ 1.2.1 无操作系统的计算机系统
- ❖ 1.2.2 单道批处理系统
- ❖ 1.2.3 多道批处理系统\*\*
- ❖ 1.2.4 分时系统\*\*
- ❖ 1.2.5 实时系统\*\*
- ❖ 1.2.6 微机操作系统
- ❖ 1.2.7 多处理机操作系统
- ❖ 1.2.8 网络操作系统
- ❖ 1.2.9 分布式操作系统
- ❖ 1.2.10 嵌入式操作系统

多道批处理系统、分时系统、实时系统并称操作系统三大基本类型





## 1.2.1 无操作系统的计算机系统

### ❖ 人工操作方式（1946-50年代（电子管））

- \* 工作方式：用户既是程序员又是操作员，或者是计算机专业人员；编程语言为**机器语言**；输入、输出为纸带或卡片。
- \* 工作特点（缺点）：**用户独占全机；CPU等待人工操作，CPU利用率低**（计算前，手工装入纸带或卡片；计算完成后，手工卸取纸带或卡片）。
- \* 两对矛盾：人工操作的低速性与CPU运算的高速性矛盾（**人机矛盾**）、CPU与I/O设备的速度不匹配矛盾。

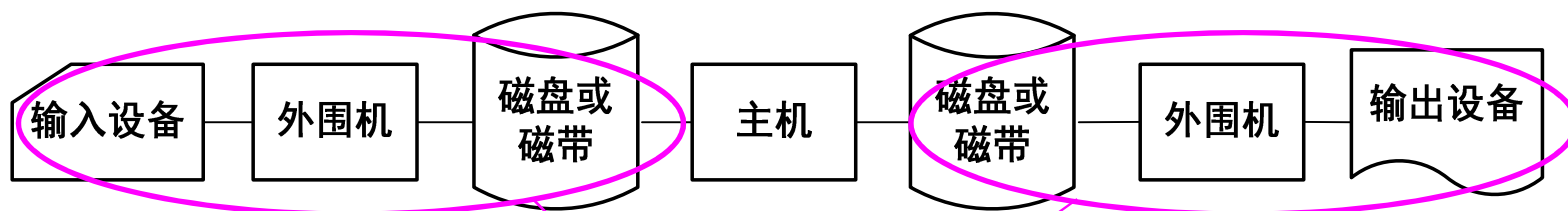




## 1.2.1 无操作系统的计算机系统

### ❖ 脱机I/O方式（Off-Line I/O）（20世纪50年代末）

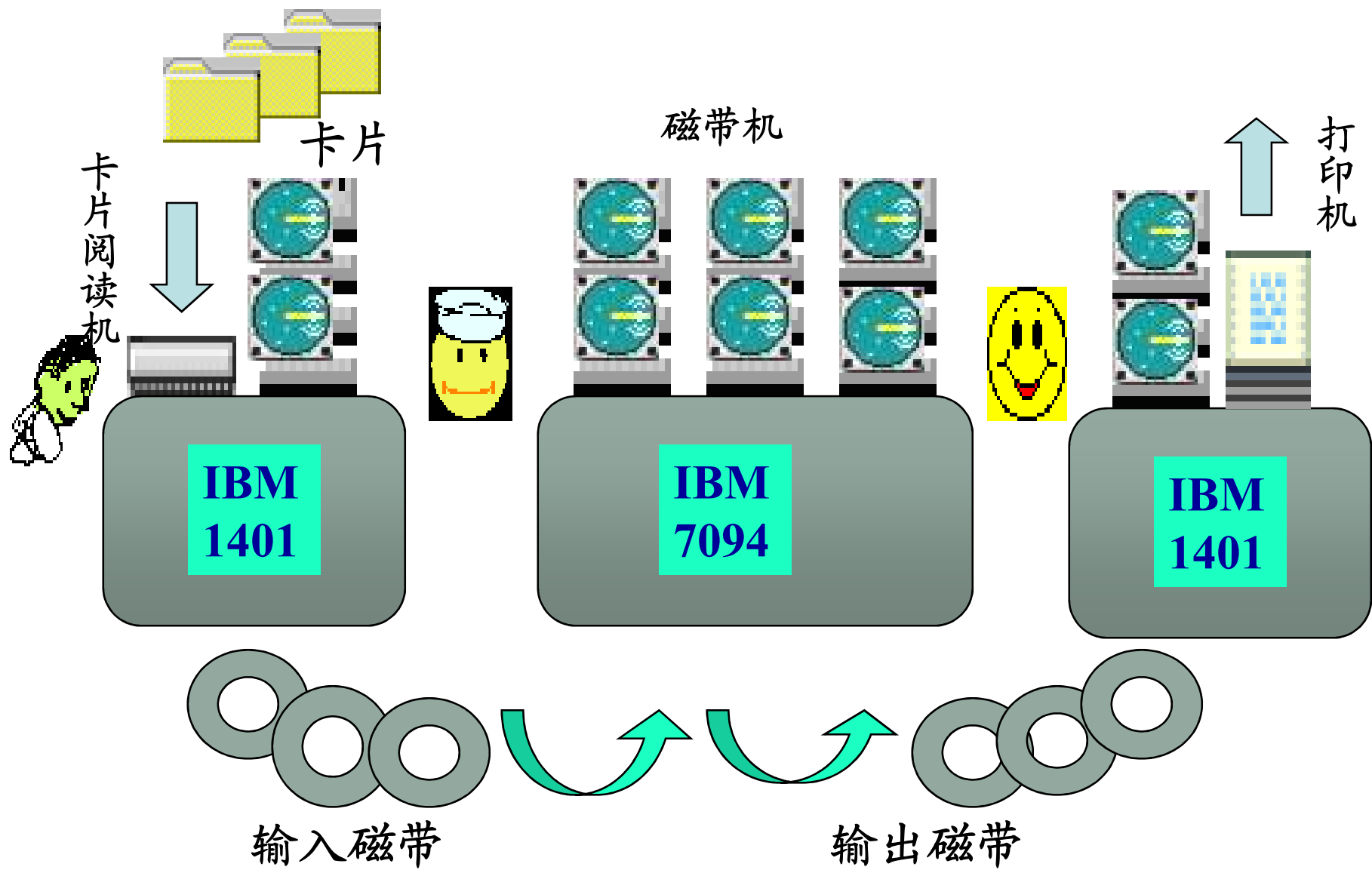
- \* 引入外围机（也叫卫星机，是一种专门用来管理I/O的、功能较简单的计算机）的概念，解决人工操作方式的缺点。
- \* 事先将用户程序或数据的纸带或卡片放入纸带输入机，在外围机的控制下输入到磁带上，CPU从磁带上读入数据；输出过程正好相反。



- \* 优点：减少了CPU的空闲时间、提高了I/O速度。

脱离主机

I/O



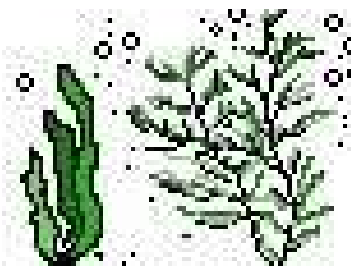
### 图1-2 脱机I/O方式



## 1.2.2 单道批处理系统

### ❖ 批处理相关概念理解

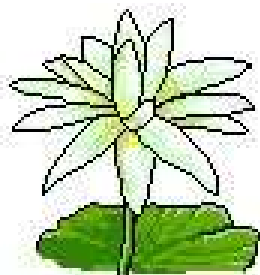
- \* 批处理中的作业的组成：用户程序+数据+作业说明书（采用作业控制语言编写）。
- \* 一批作业：通常由若干个同类型作业组装成，在处理中使用一组相同的系统软件（系统带）。
- \* 批处理技术：是指计算机系统对一批作业自动进行处理的一种技术。
- \* 批处理系统：是指加载在计算机上的一个系统软件，在它的控制下，计算机能够自动地、成批地处理一个或多个用户的作业。





## 1.2.2 单道批处理系统

- \* **中断**：是指当主机接到外界发来的中断信号时，停止原来的工作，转去处理中断的事件。在处理中断完成以后，主机又回到原来的工作点继续工作。这样可以使用户程序的I/O申请完成后，主机能自动在原中断点之后继续运行，同时**为多道程序并发执行打下了基础**。





## 1.2.2 单道批处理系统

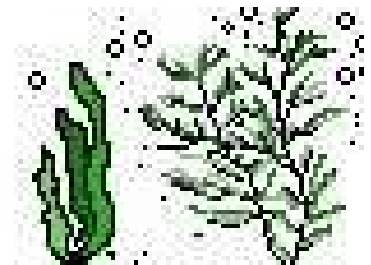
操作系统的  
雏形

### ❖ 处理过程

- \* 基本情况：1950年代末-60年代中（晶体管），利用磁带把若干个作业分类编成作业执行序列，每一批作业由专门的监督程序（Monitor）依次自动载入内存处理。可使用汇编语言开发软件。
- \* 概念：系统对作业的处理都是成批进行的、且内存中始终只保持一道作业，称为单道批处理系统（simple batch system）。
- \* 目的：单道批处理系统的引入是为了提高系统资源的利用率和吞吐量。

### ❖ 特征

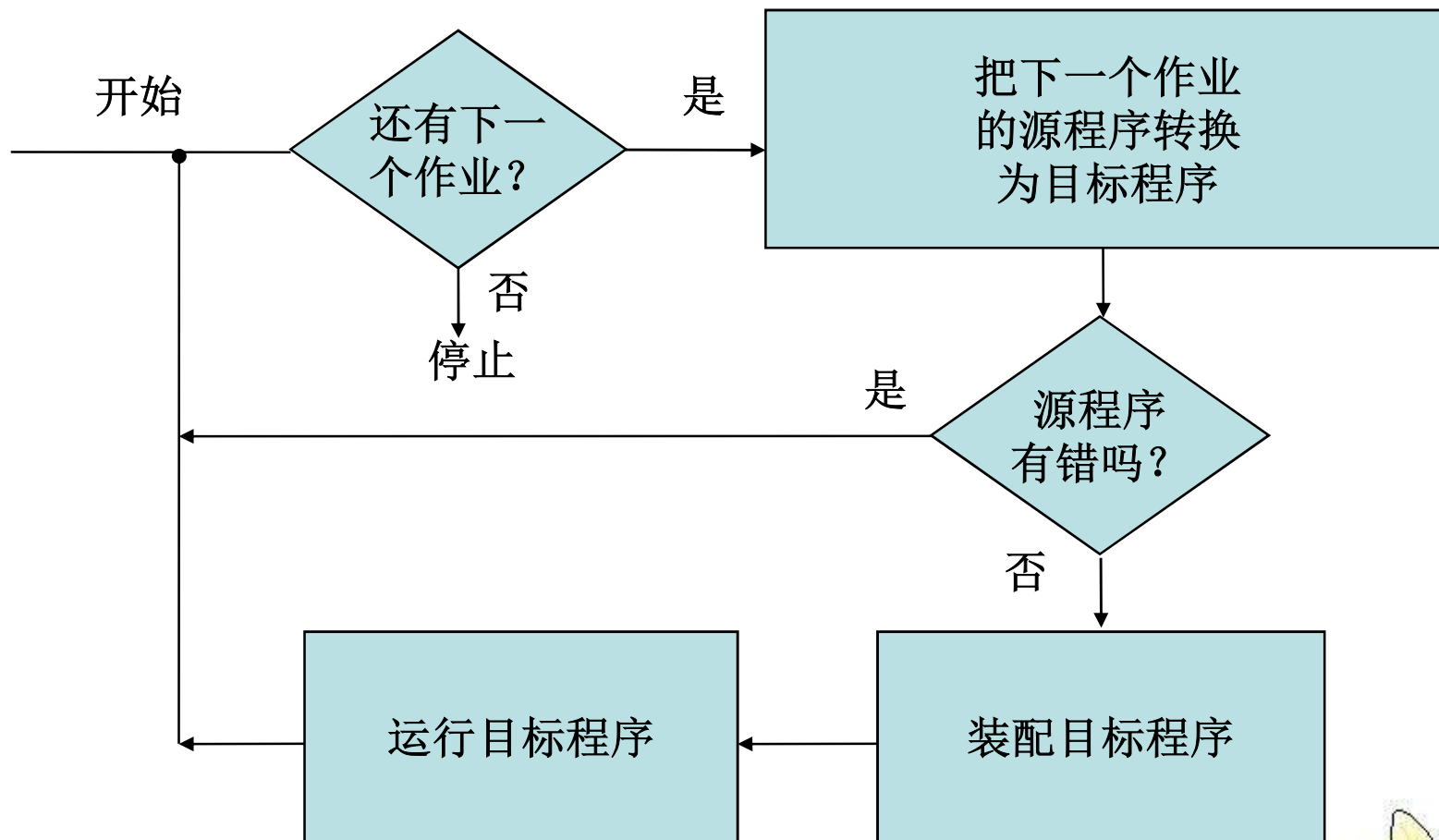
- \* 自动性、顺序性、单道性







## 图1-3 单道批处理系统的处理流程

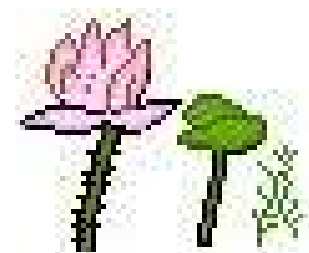




## 1.2.3 多道批处理系统

### ❖ 基本概念

- \* 时间：20世纪60年代中期（小规模集成电路）——70年代中期
- \* 多道：内存中同时驻留多个作业
- \* OS中引入多道技术的**优点**：
  - 提高CPU利用率（课件图1.4）
  - 提高内存和I/O设备利用率
  - 增加系统吞吐量



### ❖ 特征

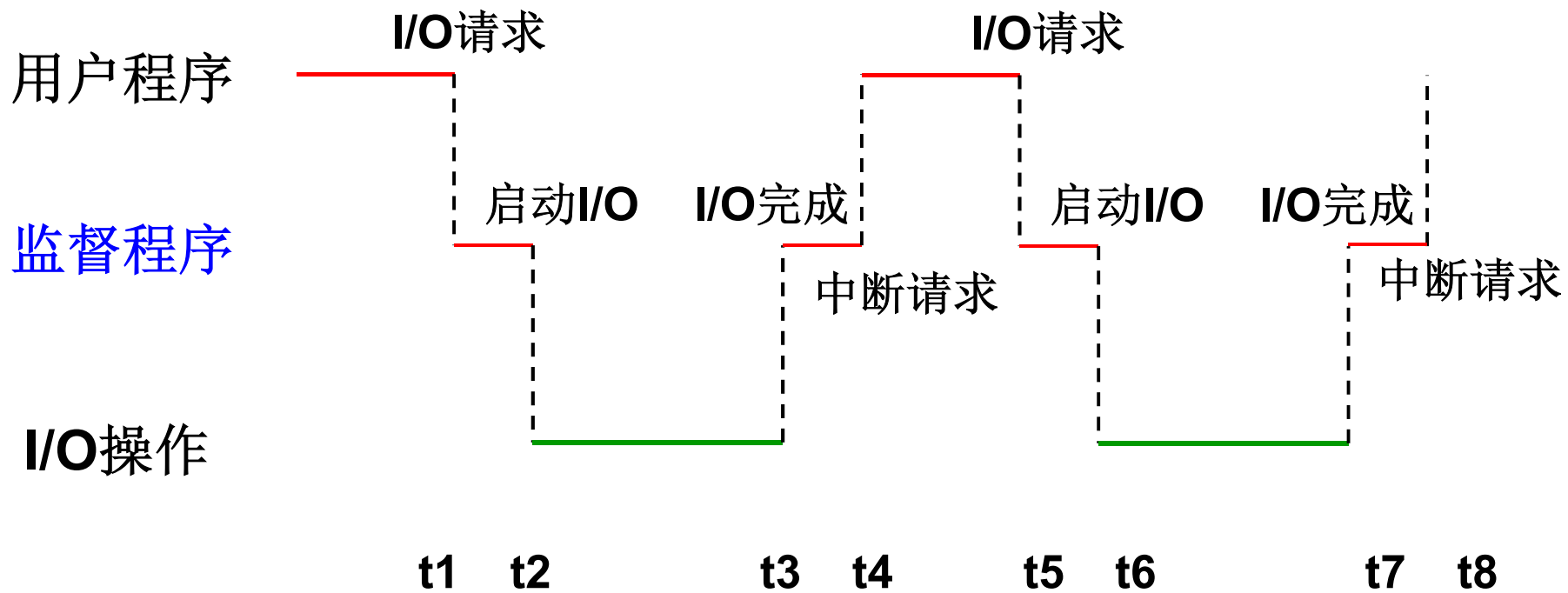
- \* **多道性、无序性**（作业调度次序与作业在外存中的次序无关、作业的完成次序与作业进入内存的顺序无关）、**调度性**（作业调度、进程调度??）

### ❖ 缺点

- \* 平均**周转时间长**（几小时、几天）、**无交互能力**



图1 - 4

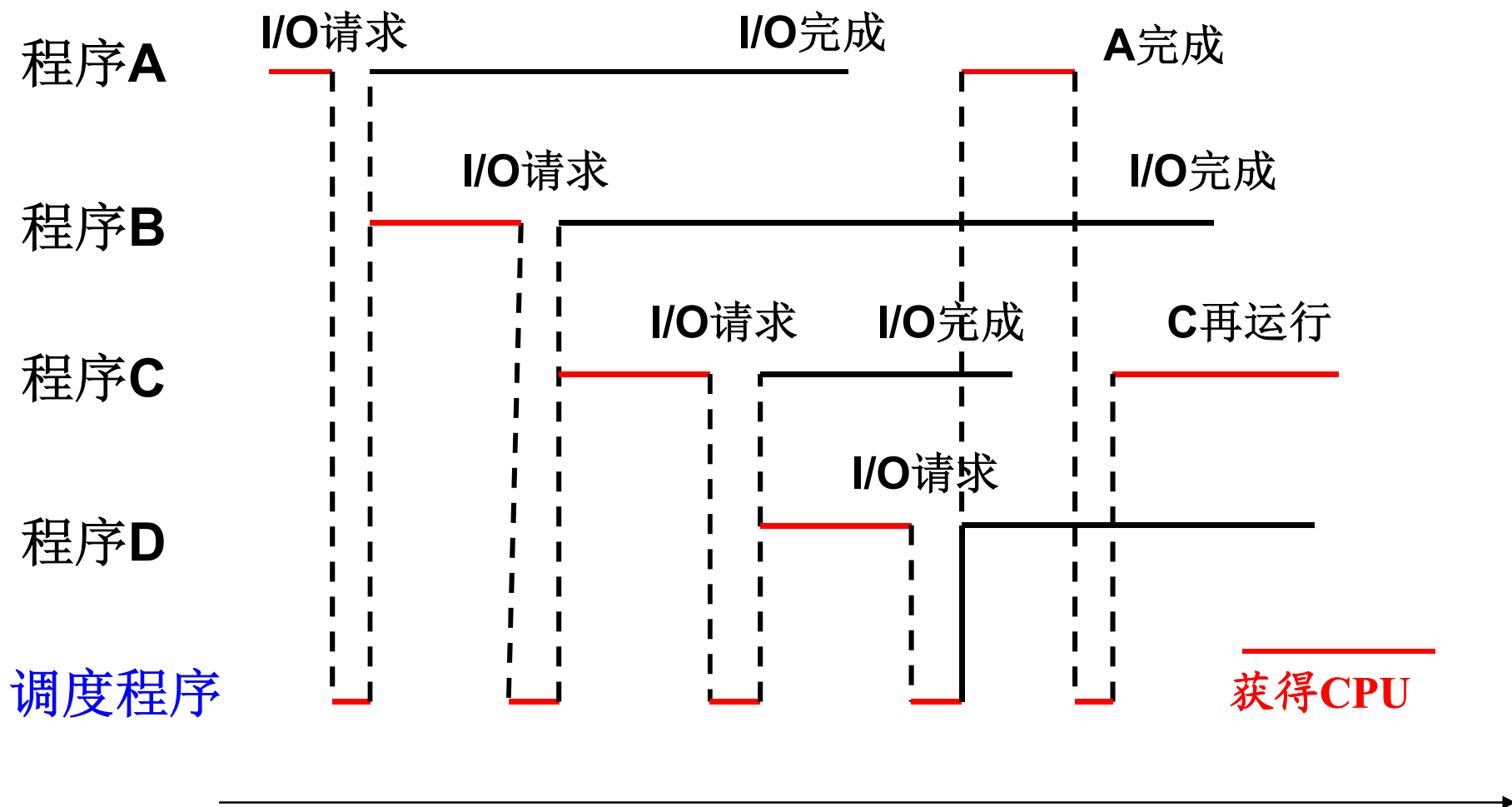


(a) 单道程序运行情况

— 获得CPU  
— CPU空闲



图1 - 4



(b) 四道程序运行情况



## 例题1-1

【例1-1】 有三个程序A、B、C,它们使用同一个设备进行I/O操作,并按A、B、C的优先次序执行。这三个程序的计算和I/O操作时间如下表所示。假设调度的时间可以忽略不计,请分别画出单道程序环境 and 多道程序环境下(假设内存中可同时装入这三道程序)它们运行的时间关系图,并比较它们的总运行时间。

程序运行时间表

单位: ms

操作	程序A	程序B	程序C
计算	30	60	20
I/O	40	30	40
计算	10	10	20



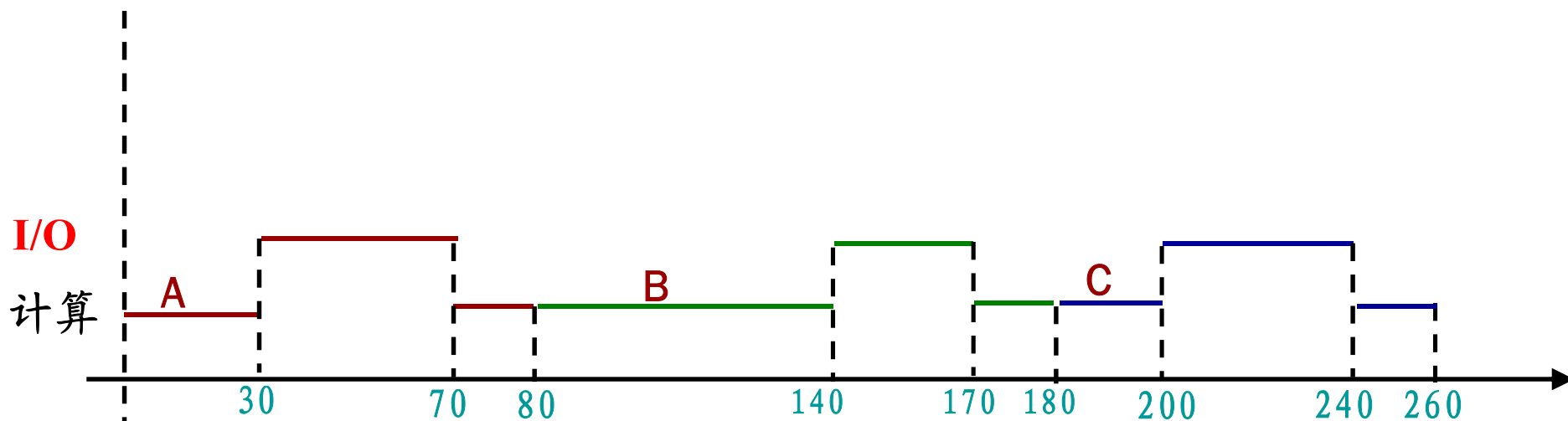
## 例题1-1

【解】 1> 单道程序环境总运行时间: **260ms**

2> 多道程序环境总运行时间

a、非抢占式 **180ms**

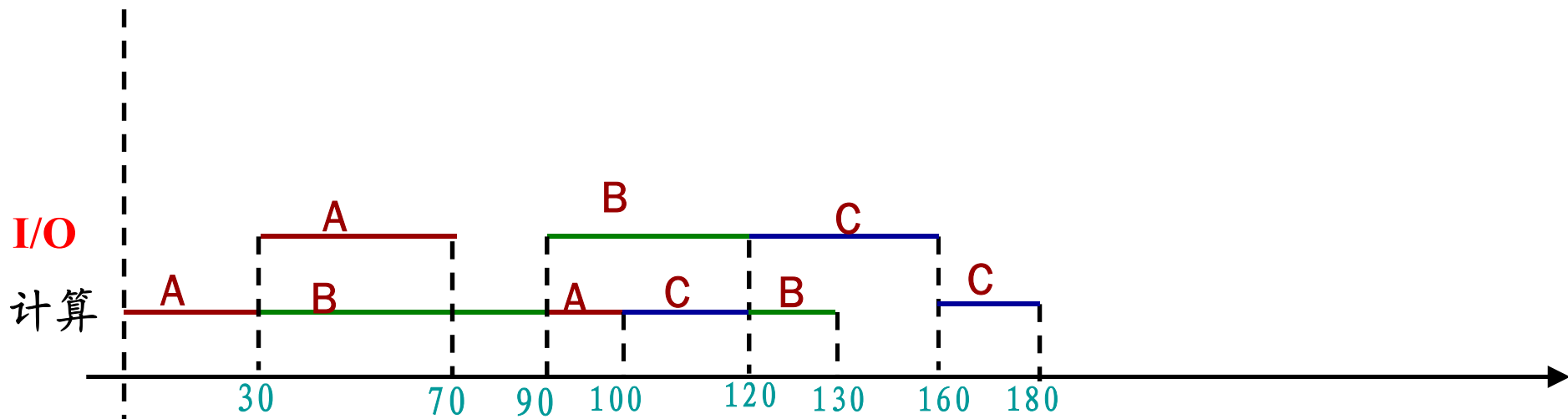
b、抢占式 (详参第三章) **190ms**



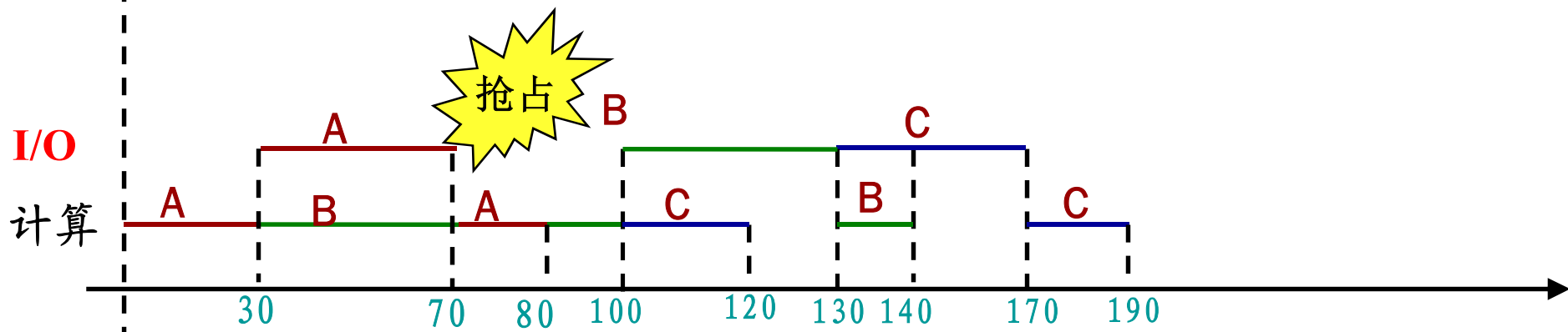
单道运行的时间关系图



# 例题1-1



多道、非抢占式运行的时间关系图



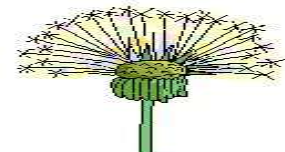
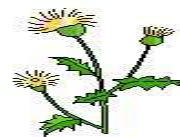
多道、抢占式运行的时间关系图



## 1.2.3 多道批处理系统

### ❖ 需解决的问题

- \* 处理机管理问题
- \* 内存管理问题
- \* I/O管理问题
- \* 文件管理问题
- \* 作业管理问题





## 1.2.3 多道批处理系统



1999 图灵奖得主  
Frederick Brooks  
1931-

### ❖ OS/360

- \* 1964年，IBM发布OS/360，由Brooks领导，共花费5000个人年、5亿美金进行开发；
- \* OS/360支持多道程序，最多可同时运行15道程序。为了便于管理，OS/360把中央存储器划分为多个（最多15个）分区，每个程序在一个分区中运行。



## 1.2.3 多道批处理系统

### ❖ 单道、多道批处理系统比较

	单道批处理系统	多道批处理系统
内存使用	每次一个作业	每次多个作业 (充分利用内存)
作业次序	顺序, 先进先出	无确定次序
占用CPU	独占	交替占用
是否需要调度	不需要	需要



## 1.2.4 分时系统

### ❖ 分时系统的产生（20世纪70年代中至今）

- \* 概念：分时系统（Time Sharing System）指一台主机上连接了多个带有显示器和键盘的终端，各用户都可通过自己的终端以交互方式使用主机，分时共享主机中的软硬件资源。
- \* 用户的需求是分时系统产生的动力：
  - 人机交互性
  - 共享主机
  - 便于用户上机





## 1.2.4 分时系统

### ❖ CTSS

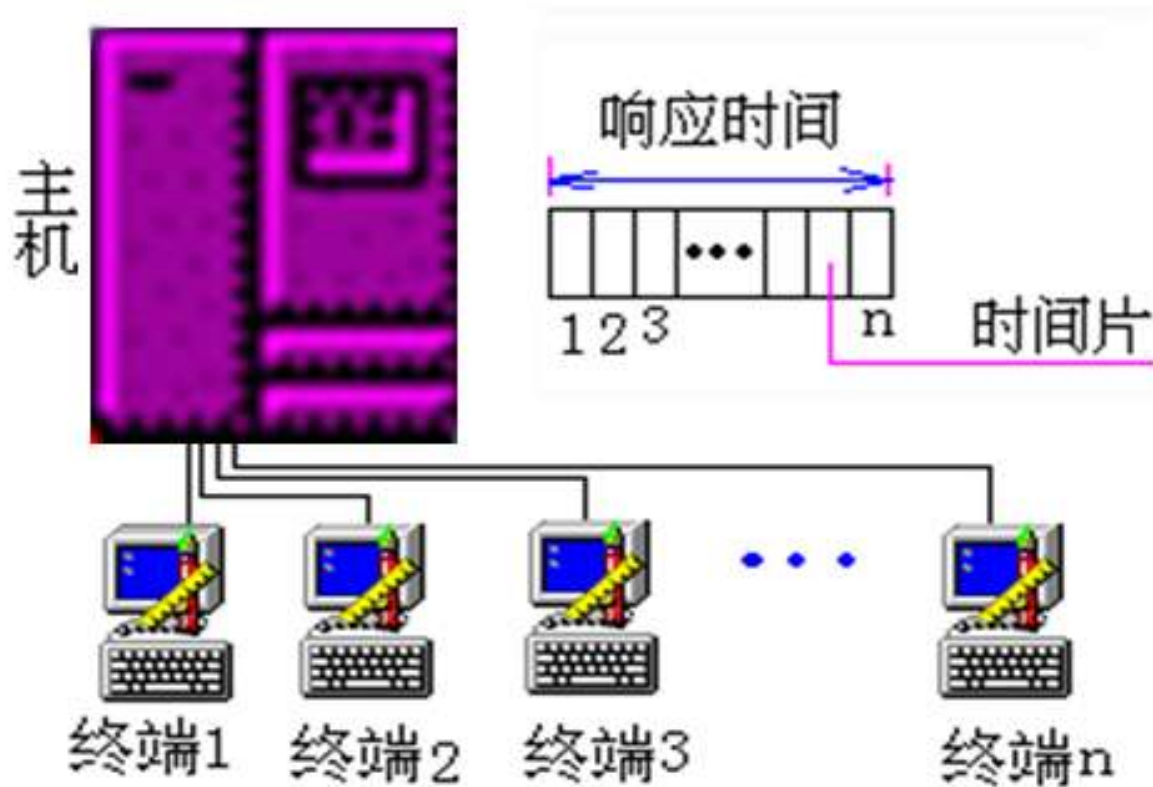


**1990 图灵奖得主**  
**Fernando José Corbató**  
**1926-**

- \* 1961年，由考巴脱领导，MIT开发成功世界上第一个分时操作系统CTSS (Compatible Time Sharing System)



## 图1-5 分时系统示意图





## 1.2.4 分时系统

### ❖ 分时系统在实现中需解决的关键问题

\* **关键问题：**使用户能与自己的作业进行交互

\* **解决方法**

- **及时接收：**多终端卡（多路卡, 用于实现分时多路复用）、为每个终端配置缓冲区
- **及时处理：**交互的作业应存放在内存、响应时间应短（主机分时处理作业，不允许一个作业长期占用处理机）





## 1.2.4 分时系统

### ❖ 分时系统类型

#### \* 单道分时系统

- 内存中的作业执行完一个时间片便调出，再调另一作业进入内存，再执行一个时间片，以此类推。

#### \* 具有前、后台的分时系统

- 内存被固定的划分成“前台区”和“后台区”两部分，“前台区”存放按时间片“调进”和“调出”的作业流，“后台区”存放批处理作业或对处理时间要求不强的作业。

#### \* 多道分时系统

- 不需要花费大量的作业调入、调出开销。
- **MULTICS (Ken Thompson UNIX之父)**







## 1.2.4 分时系统

### ❖ 分时系统的特征

- \* 多路性（同时性）：多终端共享资源
- \* 独立性：各终端彼此独立，互不干扰
- \* 及时性：用户请求响应时应短（1-3秒）
- \* 交互性：广泛性的交互





## 1.2.5 实时系统

### ❖ 实时系统及其类型

#### \* 产生背景

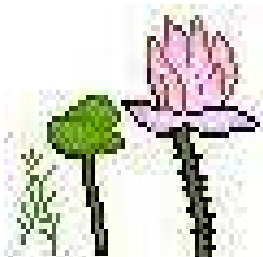
- 要求对任务及时处理

#### \* 概念

- 实时系统（Real Time System）能及时响应外部事件的请求，在规定时间内完成对该事件的处理，并控制所有实时任务协调一致地运行。

#### \* 实时系统应用类型

- **实时控制：**如导弹拦截系统、工业控制系统
- **实时信息处理：**股票交易系统
- **快速响应：**搜索引擎





## 1.2.5 实时系统

### ❖ 实时任务类型

#### \* 按任务执行是否呈现周期性来划分

- 周期性实时任务 P10
- 非周期性实时任务: 开始截止时间、完成截止时间

#### \* 根据对截止时间的要求来划分

- 硬实时任务 ( Hard real-time task ) P10
- 软实时任务 ( Soft real-time task )



## 1.2.5 实时系统

### ❖ 实时系统与分时系统特征比较

#### \* (1) 多路性

- 实时系统: 对多个对象的信息采集和控制。
- 分时系统: 为多个终端用户服务。

#### \* (2) 独立性

- 实时系统: 各个对象互不干扰。
- 分时系统: 各用户独立操作, 互不干扰。

#### \* (3) 及时性

- 实时系统: 在更短的时间内对外部事件响应。
- 分时系统: 在一定时间内处理各终端作业。



## 1.2.5 实时系统

### \* (4) 交互性

- 实时系统：交互性比较少。
- 分时系统：用户可与主机进行广泛交互。

### \* (5) 可靠性

- 实时系统要求更高。



## 1.2.6 微机操作系统

微机操作系统是指配置在微型机上的操作系统，按运行方式可分为三类：

### ❖ (1) 单用户单任务操作系统

\* 含义：这类OS只允许一个用户上机，且只允许用户程序作为一个任务运行。

\* 代表性OS：

- CP/M（20世纪70年代 8位微机）
- MS-DOS 1.0-3.3（20世纪80年代-90年代初 16位微机）





## 1.2.6 微机操作系统

### ❖ (2) 单用户多任务操作系统

- \* 含义：这类OS只允许一个用户上机，但允许用户把一个程序分为若干个任务或者启动若干程序，使它们并发执行，从而有效地改善了系统的性能。
- \* 目前在32位微机上配置的操作系统基本上都是单用户多任务操作系统，其中最具有代表性的是由微软公司推出的Windows。



## 1.2.6 微机操作系统

### ❖ (3) 多用户多任务操作系统

\* 含义：这类OS允许多个用户通过各自的终端使用同一台机器，共享主机系统中的各种资源，而每个用户可进一步启动若干个任务，使它们能并发执行，从而可进一步提高资源利用率和系统吞吐量。

\* 在大、中和小型机中所配置的大多是多用户多任务操作系统，而在32位微机上也有不少是配置的多用户多任务操作系统，其中最具有代表性的是UNIX。

\* 当前微机上最有影响的两种Unix: Solaris和Linux。





## 1.2.7 多处理机操作系统

### ❖ 多处理机系统（Multi-processor OS）

- \* 多处理机操作系统的出现是为了提高计算机系统性能和可靠性。
- \* 提高性能有两条途径：提高各个组成部分的速度、增大处理的并行程度。
- \* 多处理机系统的特点：
  - 增加系统的吞吐量： $N$ 个处理器加速比达不到 $N$ 倍（额外的调度开销，算法的并行化）
  - 提高系统可靠性：故障时系统降级运行。





## 1.2.7 多处理机操作系统

### ❖ 多处理机系统的类型

#### \* 根据耦合程度划分

- **紧密耦合MPS**: 多处理机通过高速线路互连, 它们共享主存和外设。
- **松散耦合MPS**: 通过通道或通信线路实现多台计算机互连, 每台计算机可独立工作, 必要时通过通信线路与其它计算机交互。



#### \* 根据结构是否对称划分

- **非对称MPS**: 主从模式, 主处理机只有一个, 配置操作系统, 从处理机可有多。
- **对称MPS**: 所有处理机都相同。操作系统可根据需要运行于其中的任一处理机上。



## 1.2.7 多处理机操作系统

### 中国研制成功每秒运算逾千万亿次超级计算机

天河一号

全系统峰值性能	1206万亿次/秒
Linpack实测性能	563.1万亿次/秒
共享存储总容量	1PB
全系统包含通用处理器(CPU)	6144个
全系统包含加速处理器(GPU)	5120个
互联通信网络的单根线传输速率	10Gbps
目前投资	6亿人民币
使用寿命预计	10年
全系统运行情况下耗电	1280度/小时



王永卓 编制 新华社发

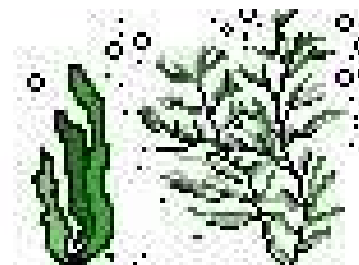
2010年11月 运算速度全球第一 国防科大研制



## 1.2.8 网络操作系统

### ❖ 基本概念

- \* 网络操作系统(NOS, Network Operating System)是在一般操作系统功能的基础上提供网络通信和网络服务功能的操作系统。网络操作系统是指使网络上各计算机能方便有效地共享网络资源、为网络用户提供各种服务的软件与有关规程的集合。
- \* 代表性网络操作系统：Intranetware、Windows2000/NT。



## 1.2.8 网络操作系统

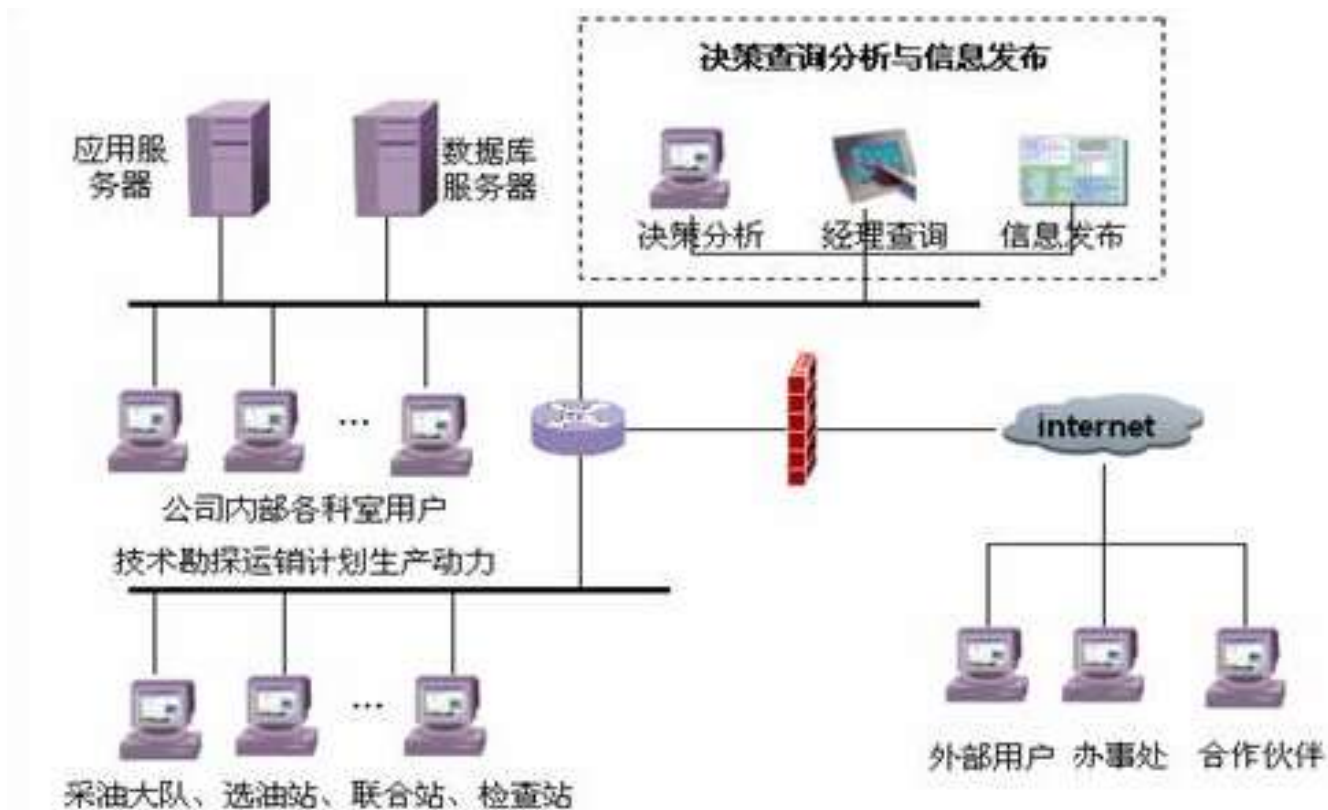


图1-6 网络操作系统示意图



## 1.2.8 网络操作系统

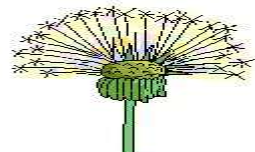
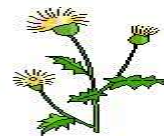
### ❖ 网络操作系统的模式

#### \* 客户机/服务器模式 (Client/Server)

- **服务器**: 为客户机提供服务, 如文件服务、打印服务、数据库服务等, 通常具有较强的处理能力。
- **客户机**: 用户通过客户机访问服务器, 自身也具有一定的处理能力。
- **连接支持部分**: 网络

#### \* 对等模式 (Peer-to-Peer)

- 各机器是对等的, 它既可作为客户机也可作为服务器。

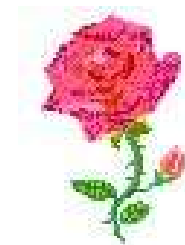




## 1.2.9 分布式操作系统

### ❖ 基本概念

- \* 分布式系统：是指由多个分散的处理单元经网络的连接而形成的系统。系统的处理和控制功能都分散在系统的各个处理单元上，系统中的所有任务可以动态地分配到各个处理单元中去。
- \* 分布式系统是以计算机网络为基础的，它的基本特征是处理上的分布，即功能和任务的分布。







## 1.2.9 分布式操作系统

### ❖ 分布式操作系统与网络操作系统的比较

#### \* (1) 分布性

- **分布式系统**: 只有一个分布式操作系统;
- **网络系统**: 各个结点可以有不同的网络操作系统。

#### \* (2) 并行性

- **分布式系统**: 可以将一个或多个任务动态分配到不同的单元上并行处理;
- **网络系统**: 每个用户的一个或多个任务通常都在本地处理。





## 1.2.9 分布式操作系统

### \* (3) 透明性

- **分布式系统**：具有透明性，如用户要访问某文件，他只需要知道文件名，无需知道文件在哪个站点。
- **网络系统**：用户要访问一个文件时要知道文件名及位置。

### \* (4) 共享性

- **分布式系统**：各站点的所有资源都可供全系统共享。
- **网络系统**：一般仅有服务器上的部分资源可供全网共享。





## 1.2.9 分布式操作系统

### \* (5) 健壮性

- **分布式系统**: 有健壮性, 当某站点出现故障时, 正在该站点处理的任务能被自动迁移到好的站点。
- **网络系统**: 若服务器故障往往导致全网瘫痪。

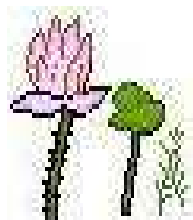




## 1.2.10 嵌入式操作系统

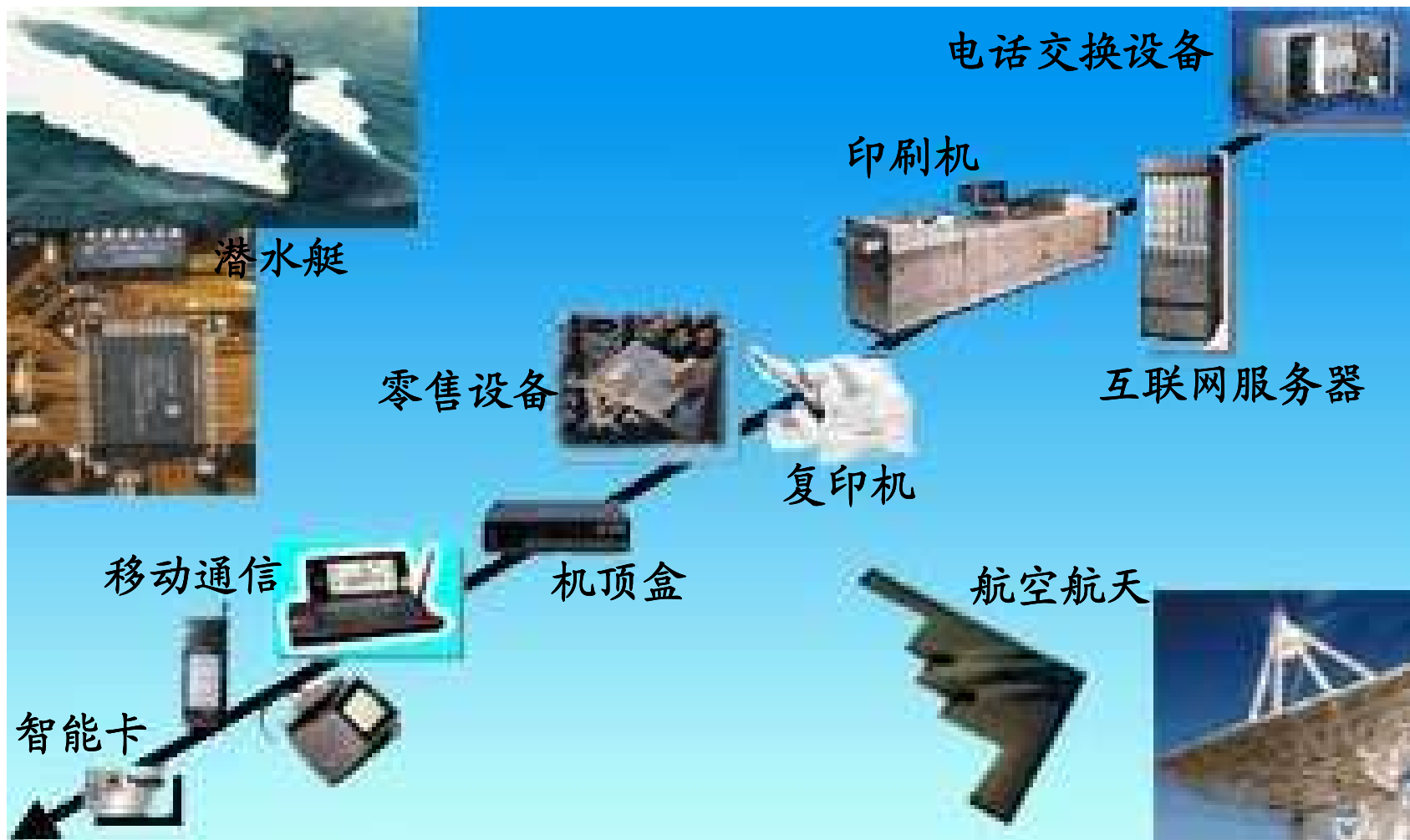
### ❖ 什么是嵌入式系统？

- \* 在各种设备、装置或系统中，完成特定功能的**软硬件系统**；
- \* 它们是一个大设备、装置或系统中的一部分，这个大设备、装置或系统可以不是“计算机”；
- \* 通常工作在反应快或对处理时间有较严格要求的环境中；
- \* 由于它们被嵌入在各种设备、装置或系统中，因此称为**嵌入式系统**。





# 图1-7 嵌入式系统的应用

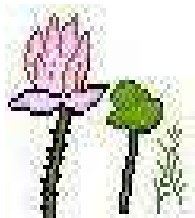




## 1.2.10 嵌入式操作系统

### ❖ 嵌入式操作系统

- \* **概念：**嵌入式OS是运行在嵌入式智能芯片环境中，对整个智能芯片以及它所操作、控制的各部件装置等资源进行统一协调、调度、指挥和控制的系统软件。
- \* **当前流行智能手机操作系统：iPhone OS、Google Android、Windows Mobile、Symbian等。**





## 附：Operating System Market Share (2011/01)





## 附：Operating System Market Share (2011/01)

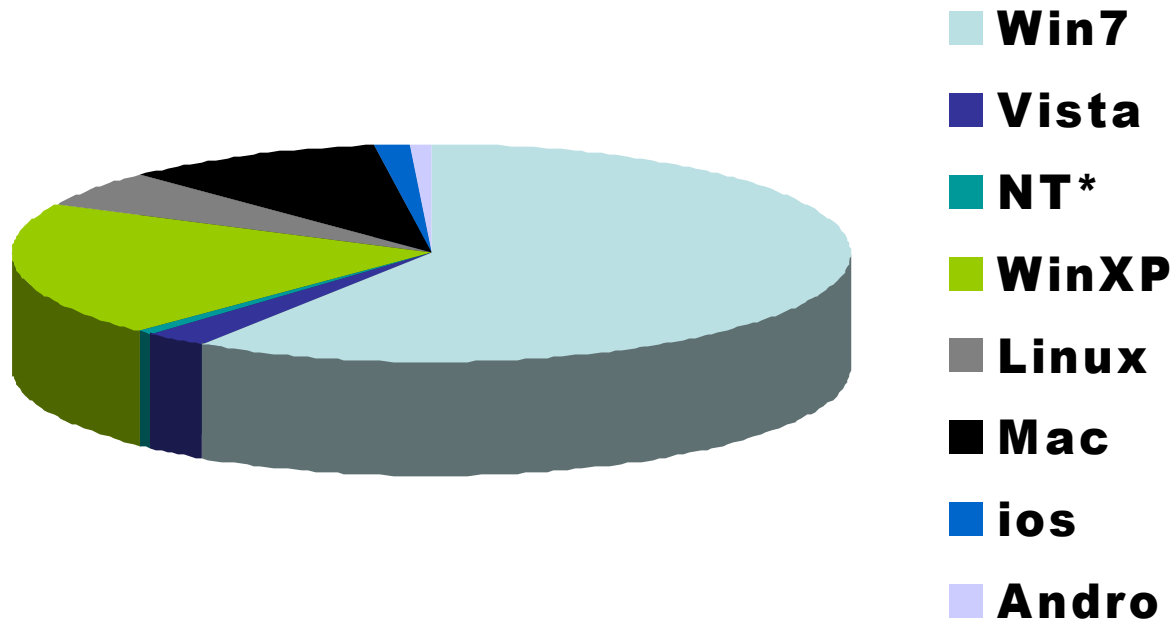
Operating System	Total Market Share
Windows	89.70%
Mac	5.25%
iOS	2.05%
Linux	0.95%
Java ME	0.81%
Android	0.49%
Symbian	0.28%
BlackBerry	0.12%
Windows Mobile	0.03%
Playstation	0.03%
FreeBSD	0.01%
BREW	0.01%
Nintendo Wii	0.00%
SunOS	0.00%
Nintendo DS	0.00%
Palm	0.00%
Web TV	0.00%
OpenBSD	0.00%
NetBSD	0.00%
HP-UX	0.00%
SCP	0.00%
AIX	0.00%
SCO	0.00%



## 附：Operating System Market Share (2013/01-02)



2013	Win8	Win7	Vista	NT*	WinXP	Linux	Mac	ios	Andro
2月	5.7	55.3	2.4	0.5	19.1	4.8	9.6	1.06	0.91
1月	4.8	55.3	2.6	0.5	19.9	4.8	9.3	1.07	0.90

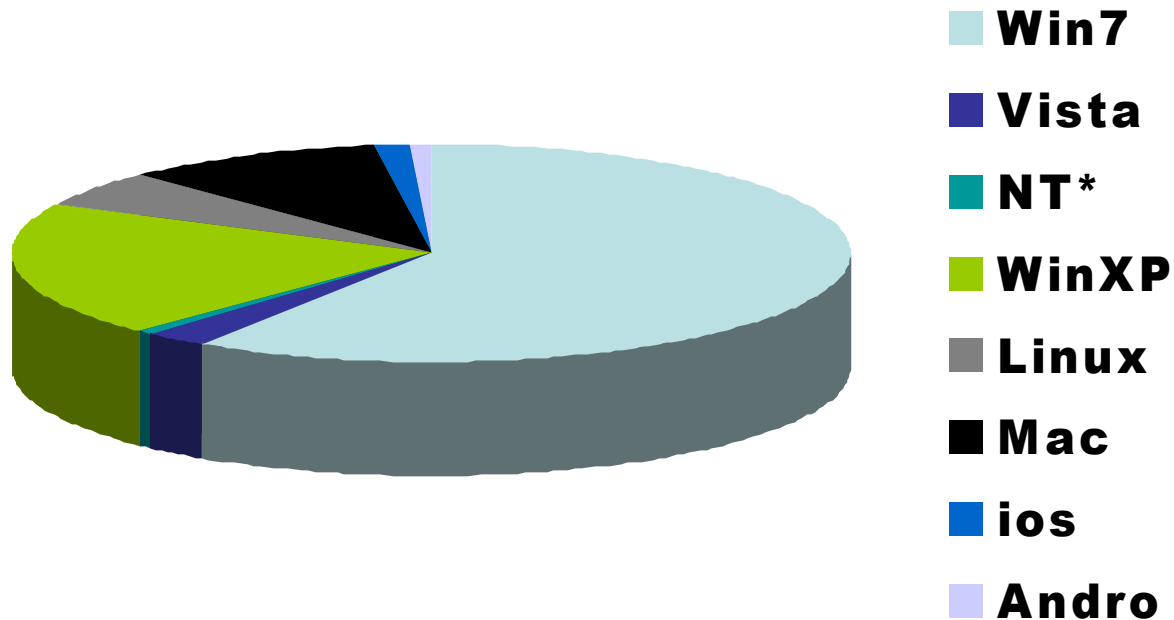




## 附：Operating System Market Share (2014/06-07)



2014	Win8	Win7	Vista	NT*	WinXP	Linux	Mac	ios	Andro
7月	17.3	54.8	1.0	0.2	7.0	5.6	9.5	1.13	2.46
6月	17.0	55.3	1.1	0.2	7.1	5.3	9.6	1.23	2.31



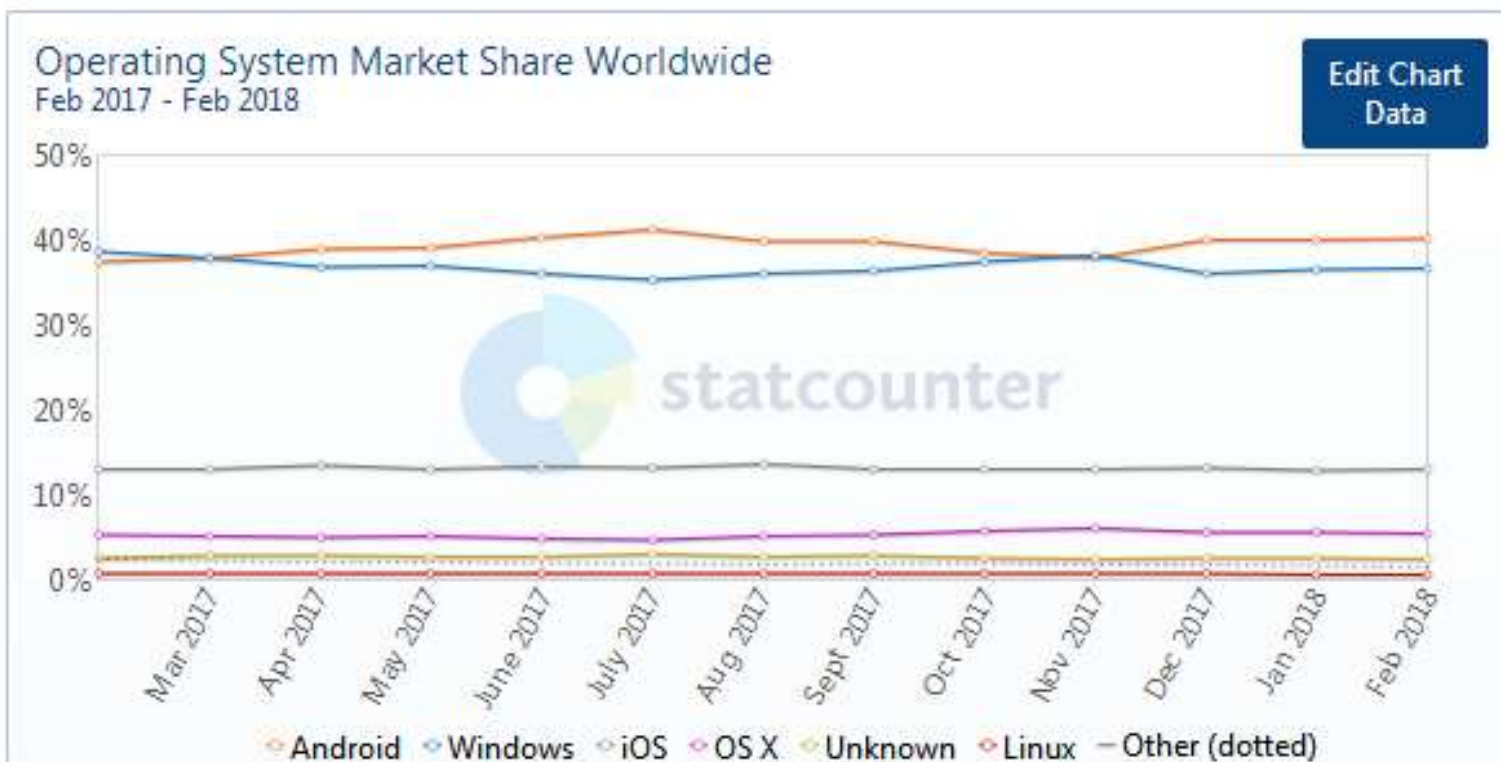




# 附：Operating System Market Share (2018/02)

Android	Windows	iOS	OS X	Unknown	Linux
40.24%	36.58%	13.08%	5.43%	2.46%	0.67%

Operating System Market Share Worldwide - February 2018

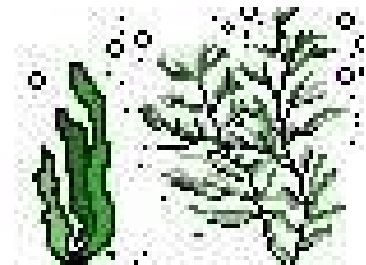




## 1.3 操作系统的基本特性

- ❖ 1.3.1 并发性 (Concurrence)
- ❖ 1.3.2 共享性 (Sharing)
- ❖ 1.3.3 虚拟技术 (Virtual)
- ❖ 1.3.4 异步性 (Asynchronism)

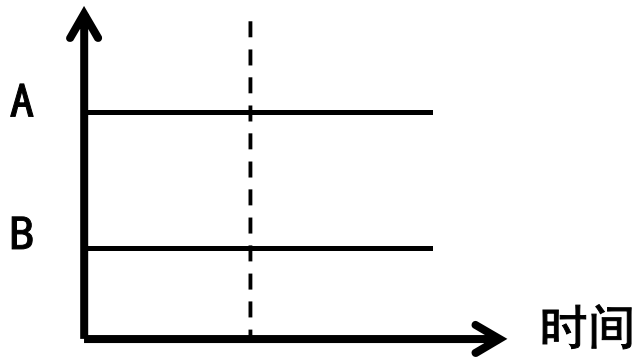
OS的两个最基本的特性



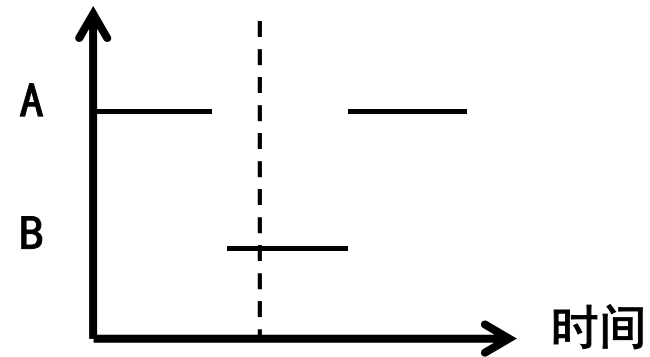


## 1.3.1 并行性

### ❖ 并行与并发



并行性是指两个或多个事件在同一时刻发生。



并发性是指两个或多个事件在同一时间间隔内发生。

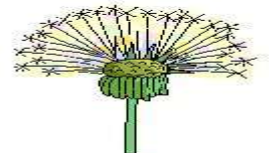
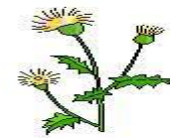




## 1.3.1 并行性

\* 在多道程序环境下多个程序如何并发运行？

- 宏观上看：多个程序是在“同时”运行；
- 微观上看：对单处理机环境，这些程序其实是分时交替即并发运行；对多处理机环境，这些程序如被分配到不同处理机，则可并行运行（真正地同时运行）。

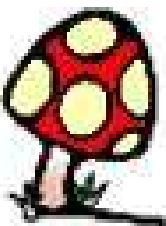




## 1.3.1 并行性

### ❖ 进程与线程

- \* 程序是静态实体，并不能自行运行，在多道程序系统中更不能和其它程序并发执行。进程就是OS为了使多个程序能并发执行而引入的。
- \* 进程是程序的一次执行，是进程实体（程序段+相关数据段+进程控制块）的运行过程，是系统进行资源分配和调度的一个独立单位，是一个能独立运行的活动实体。但由于其拥有自己的资源，在调度过程时开销太大，于是在20世纪80年代中期提出了线程。
- \* 在多线程OS中，将资源的基本单位与调度和分配的基本单位分开处理，一个进程可以含有多个线程，进程只是拥有资源的基本单位，线程才是能独立运行、调度和分配的基本单位。





## 1.3.2 共享性

### ❖ 概念

- \* 在操作系统环境下，共享是指系统中的资源可供内存中多个并发执行的进程(线程)共同使用。

### ❖ 资源共享的两种方式

- \* 互斥共享：一段时间内只允许一个进程访问某资源（称为临界资源或独占资源，如打印机等物理设备、某些软件中的栈、变量和表格）。
- \* 同时访问：一段时间内允许多个进程“同时”访问某资源（如磁盘设备、可重入码）。对单处理机环境，同时访问微观上仍是“互斥”的，是利用分时交替来实现并发访问的。





## 1.3.3 虚拟技术

### ❖ 概念

- \* 操作系统中的所谓“**虚拟**”，是指通过某种技术把一个物理实体变为若干个逻辑上的对应物。用于实现虚拟的技术，称为**虚拟技术**。

### ❖ OS中的两种虚拟技术

- \* 时分复用技术
  - 虚拟处理机技术：CPU分时
  - 虚拟设备技术：将独占的I/O设备虚拟成多台I/O设备共享
- \* 空分复用技术
  - 虚拟磁盘技术：将硬盘划分成若干个盘（卷）
  - 虚拟存储器技术：虚拟内存（即程序分段分时复用内存）







## 1.3.4 异步性

### ❖ 异步性

- \* 多道程序并发运行时,由于受资源等因素的限制,使进程的执行通常都不是“**一气呵成**”,而是以“**停停走走**”的方式运行, 进程的运行进度不可预知, 此即进程的**异步性**。
- \* 进程的异步性会给OS的管理造成混乱,为此OS通过**进程同步机制**(指多个进(线)程并发运行的协调机制,只要运行环境不变,程序经多次运行都会获得完全相同的结果)来解决这一问题。





# 小结：操作系统各特性之间的关系

## ❖ 并发和共享是操作系统最基本的特性

- \* 为了提高计算机资源的利用率，OS必然要采用多道程序设计技术，使多个程序共享系统的资源，并发地执行。

## ❖ 并发和共享互为存在的条件

- \* 一方面，资源的共享以程序（进程）的并发执行为条件，如系统不允许程序并发执行，自然不存在资源共享问题；
- \* 另一方面，若系统不能对资源共享实施有效的管理，协调好诸进程对共享资源的访问，也将影响到程序的并发执行，甚至根本无法并发执行。



## 小结：操作系统各特性之间的关系

### ❖ 虚拟技术以并发和资源共享为前提

- \* 为了使并发进程能更方便、更有效地共享资源，操作系统常采用多种虚拟技术来在逻辑上增加CPU和设备的数量以及存储器的容量，从而解决众多并发进程对有限的系统资源的争用问题。

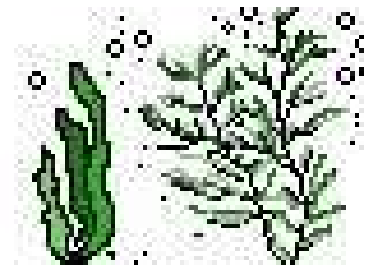
### ❖ 异步性是并发和共享的必然结果

- \* 操作系统允许进程共享资源、相互合作，使得每个进程的运行过程受到其他进程的制约，不再“一气呵成”，这必然导致异步性特征的产生。



## 1.4 操作系统的主要功能

- ❖ 1.4.1 处理机管理功能
- ❖ 1.4.2 存储器管理功能
- ❖ 1.4.3 设备管理功能
- ❖ 1.4.4 文件管理功能
- ❖ 1.4.5 操作系统和用户之间的接口
- ❖ 1.4.6 现代OS的新功能





## 1.4.1 处理机管理功能

❖ 处理机是计算机中**最重要的资源**，多道环境下，处理机的运行及分配都是**以进程为单位**，因此**处理机管理可归结为进程管理**；在引入线程的OS中，还必须包含对线程的管理。

### ❖ 管理功能

- \* 进程控制
- \* 进程同步
- \* 进程通信
- \* 调度





## 1.4.1 处理机管理功能

### ❖ 1、进程控制

- \* 创建/撤消进程
- \* 控制进程在运行过程中的状态转换
- \* 创建/撤消线程（如果OS引入了线程）

### ❖ 2、进程同步

- \* 主要任务：协调多个进（线）程的运行。
- \* 协调方式：
  - 进程互斥方式：进程以互斥方式访问临界资源。
  - 进程同步方式：进程相互合作完成共同任务，由同步机构协调它们的执行次序。



## 1.4.1 处理机管理功能

### ❖ 3、进程通信

- \* **主要任务：** 实现相互合作的进（线）程之间信息的交换/传递。
  - 例如：输入进程、计算进程、打印进程相互间有信息传递
- \* **通信方式：**
  - **直接通信：** 进程A发message，进程B收message
  - **间接通信：** 进程A发message到中间实体（如mailbox），进程B从中间实体收message



## 1.4.1 处理机管理功能

### ❖ 4、调度

- \* **作业调度**：从后备队列中按照一定算法，选择出若干个作业，为他们**建立进程，分配必需的资源（首先是分配内存）**，并按一定算法将他们插入就绪队列。（注：这句话只适于三状态OS，对含有创建状态的OS并不严谨，详参教材P37）
- \* **进程调度**：从就绪队列中按一定算法选一个进（线）程，为其**分配处理机，设置运行现场**，以使其运行。



## 1.4.2 存储器管理功能

### ❖ 管理对象

- \* 存储器管理的对象主要是内存，内存是计算机中第二重要的资源。

### ❖ 主要任务

- \* 为多道程序的运行提供环境，方便用户使用，提高存储器利用率，能从逻辑上扩充内存。

### ❖ 管理功能

- \* 内存分配
- \* 内存保护
- \* 地址映射
- \* 内存扩充







## 1.4.2 存储器管理功能

### ❖ 1、内存分配

- \* **主要任务：** 为每道程序分配内存空间；提高内存利用率，减少不可用的内存空间；允许正在运行的程序动态申请附加内存。
- \* **分配方式：**
  - **静态分配：** 内存空间是确定不变的；且不允许作业在内存中“移动”。
  - **动态分配：** 内存空间并非确定不变，作业在运行过程中可动态申请附加的内存空间；作业在内存中可“移动”。



## 1.4.2 存储器管理功能

\* 内存分配机制中应具有的功能和结构:

- 内存分配数据结构: 记录内存空间使用情况
- 内存分配功能: 给程序分配内存空间
- 内存回收功能: 回收不再用的内存空间

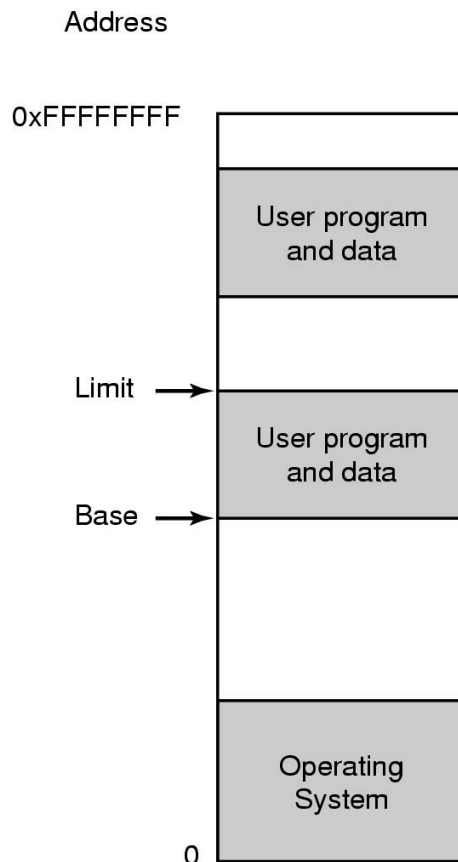




## 1.4.2 存储器管理功能

### ❖ 2、内存保护

- \* 主要任务：确保每道程序都只在自己的内存空间内运行，彼此互不干扰。
- \* 简单的内存保护机制：设置两个界限寄存器（用于存放正在执行程序的上界与下界），对每条指令进行越界检查（一般由硬件实现）

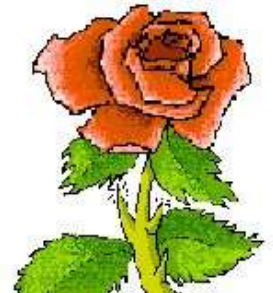




## 1.4.2 存储器管理功能

### ❖ 3、地址映射

- \* 程序中的地址称为“**逻辑地址**”或“**相对地址**”，这些程序的起始地址都是从“0”开始，程序中的其它地址都是**相对**于起始地址计算；程序中的地址所形成的范围称为“**地址空间**”。
- \* 内存中单元的地址称为“**物理地址**”；由内存中的一系列单元所限定的地址范围称为“**内存空间**”。
- \* 把地址空间的逻辑地址转换为内存空间的物理地址的过程为**地址映射**。
- \* 地址映射应在**硬件的支持**下完成。





## 1.4.2 存储器管理功能

### ❖ 4、内存扩充

\* **主要任务：**利用虚拟存储技术，**从逻辑上扩充内存容量**，以方便大作业的运行和增加内存中并发作业的道数。

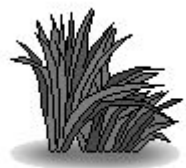
\* **实现方法：**

- **请求调入功能**

- **允许装入一部分程序和数据**就可以运行，运行时，若发现继续运行所需的程序和数据不在内存，再向OS发出请求，将其调入。

- **置换功能**

- 调出暂时不用的程序和数据，在该内存空间上调入继续运行所需的程序和数据。





## 1.4.3 设备管理功能

### ❖ 管理对象

- \* 用于管理计算机系统中所有外设，琐碎而繁杂，因为外设品种繁多、用法各异、各类外设与主机速度级差很大。

### ❖ 主要任务

- \* 完成用户进程提出的I/O请求;为用户进程分配其所需的I/O设备;提高CPU和I/O设备的利用率;提高I/O速度;方便用户使用I/O设备。

### ❖ 管理功能

- \* 缓冲管理
- \* 设备分配
- \* 设备处理





## 1.4.3 设备管理功能

### ❖ 1、缓冲管理

- \* **主要任务**：为设备提供缓冲区以**缓解CPU与外设的I/O速度不匹配的矛盾**，提高CPU的利用率，进而提高系统吞吐量。
- \* **缓冲区机制**
  - 单缓冲、双缓冲、循环缓冲、公共缓冲池





## 1.4.3 设备管理功能

### ❖ 2、设备分配

#### \* 主要任务

- 根据用户进程的I/O请求、系统现有资源的使用情况以及一定的分配策略，为其分配设备。
- 如果I/O设备与CPU之间存在设备控制器和I/O通道（P184-186），还需要为分配出去的I/O设备分配相应的控制器和通道。







## 1.4.3 设备管理功能

### ❖ 3、设备处理

- \* 设备处理程序：又称**设备驱动程序**，其基本任务是实现CPU和设备控制器或I/O设备之间的通信。
- \* 处理流程：CPU $\longleftrightarrow$ 设备驱动程序 $\longleftrightarrow$  I/O通道 $\longleftrightarrow$ 设备控制器 $\longleftrightarrow$  I/O设备



## 1.4.4 文件管理功能

### ❖ 主要任务

- \* 对外存上的用户文件和系统文件进行管理,以方便用户使用并保证文件的安全性。

### ❖ 管理功能

- \* 文件存储空间的管理
- \* 目录管理
- \* 文件的读/写管理和保护





## 1.4.4 文件管理功能

### ❖ 1、文件存储空间管理

- \* 主要任务：为每个文件分配必要的外存空间，提高外存利用率，并能有助于提高文件系统的存、取速度。
- \* 文件存储空间管理需具备的结构和功能：
  - 外存分配数据结构：记录外存空间使用情况
  - 外存分配功能：给文件分配外存空间
  - 外存回收功能：回收不再用的外存空间



## 1.4.4 文件管理功能

### ❖ 2、目录管理

- \* **主要任务**：通过目录的方式来组织文件，实现文件按名存取，提高文件的检索速度，对多用户OS，还应实现文件共享。

### ❖ 3、文件的读/写管理和保护

- \* **文件的读/写管理**：根据用户的请求，从外存中读取数据；或将数据写入外存。
- \* **文件保护（文件存取控制）**
  - 1> 防止未经核准的用户存取文件；
  - 2> 防止冒名顶替存取文件；
  - 3> 防止以不正确的方式使用文件。





## 1.4.5 操作系统和用户之间的接口

### ❖ 主要任务

- \* 便于用户（普通用户、程序员）使用。

### ❖ 接口类型

#### \* 用户接口

- 命令用户接口（由一组命令集组成）

- 联机用户接口

- » 由一组键盘操作命令及命令解释程序所组成

- 脱机用户接口（批处理用户接口）

- » 用JCL写作业说明书

- 图形用户接口

#### \* 程序接口

- 系统调用（为高级语言提供相应的库函数）





## 1.4.6 现代OS的新功能

### ❖ 1、系统安全

- \* 认证技术、密码技术、访问控制技术、反病毒技术

### ❖ 2、网络功能与服务

- \* 网络通信、资源管理、应用互操作

### ❖ 3、支持多媒体

- \* 接纳控制功能、实时调度、多媒体文件的存储



## 1.5 OS的结构设计

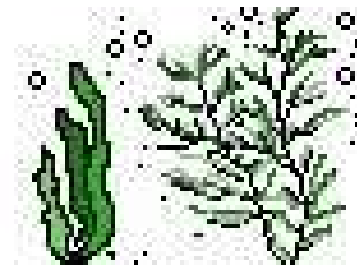
随着操作系统的发展，OS规模越来越大，代码量愈来愈多，因此必须采用良好的体系结构，采用工程化的方法进行开发。软件开发技术不断发展，促进了OS结构的更新换代。

### ❖ 1.5.1 传统OS结构

- \* 无结构（第一代）
- \* 模块化结构（第二代）
- \* 分层式结构（第三代）

### ❖ 1.5.2 现代OS结构

- \* 微内核结构（第四代）





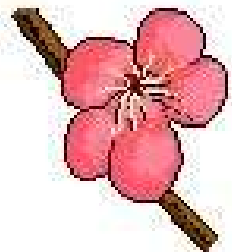
## 1.5.1 传统OS结构

### ❖ 1、无结构OS

#### \* 特点:

- 设计目标基于功能实现和获得高的效率
- OS是为数众多的一组过程的集合，各过程可相互调用
- 程序代码内部缺乏清晰的结构（goto语句非常多）
- OS整体（程序之间）无结构，也叫**整体系统结构**

#### \* 缺点：逻辑复杂，维护困难





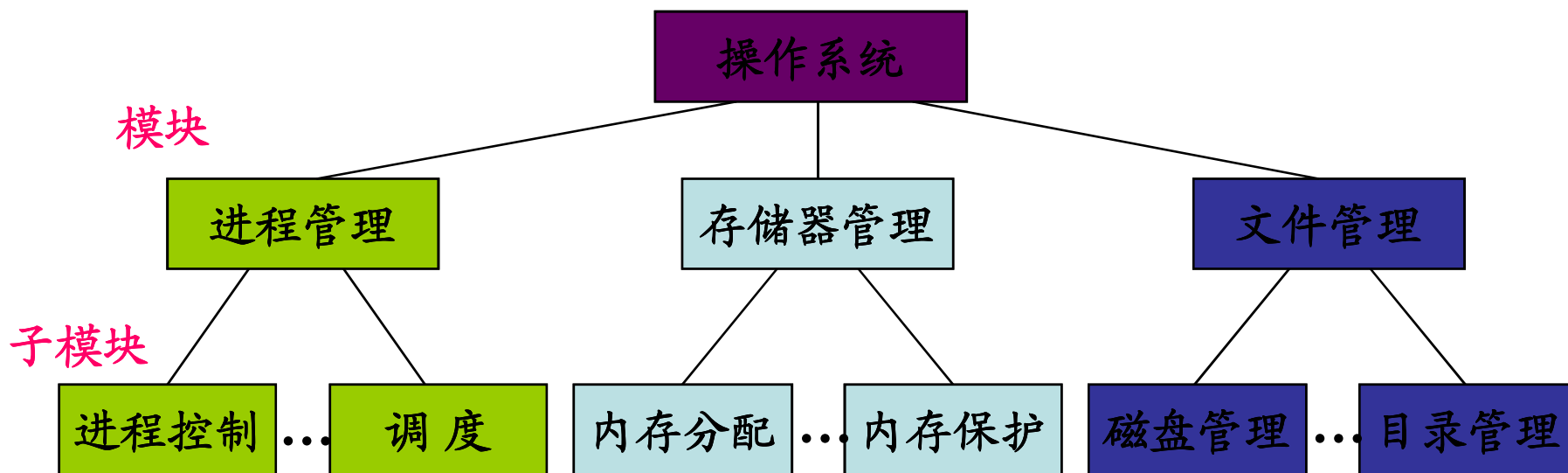


## 1.5.1 传统OS结构

### ❖ 2、模块化结构OS

#### \* 1> 模块化程序设计技术——模块接口法

- 20世纪60年代出现，基于“分解”和“模块化”原则来控制大型软件的复杂度。





## 1.5.1 传统OS结构

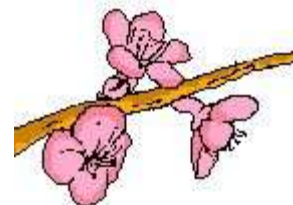
### \* 2> 模块的独立性

#### ■ 模块接口法的关键问题

- 模块大小的划分
- 模块接口的规定（为了实现模块之间的交互）

#### ■ 衡量模块独立性的指标

- **内聚性**：模块内部各部分间联系的紧密程度，越高越好。
- **耦合度**：模块间相互联系和相互影响的程度，越低越好。





## 1.5.1 传统OS结构

### \* 3> 模块接口法设计OS的优缺点

#### ■ 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的可适应性（可扩充性）
- 加快了OS的开发过程

#### ■ 缺点

- 设计时对模块的划分与接口的规定难以精确
- 模块间存在复杂的依赖关系



# 1.5.1 传统OS结构

## ❖ 3、分层式结构OS

### \* 1> 有序分层的基本概念

- 下层为上层提供服务
- 各层之间单向依赖，可简化设计的复杂度

### \* 2> 分层设计的基本原则

- 每一层都仅使用其下一层所提供的功能和服务。

### \* 3> 层次设置应考虑的因素

- 程序嵌套：各模块间的嵌套关系的复杂度
- 运行效率：随层次增高，相应软件的运行速度随之下降
- 公用模块：低层
- 用户接口：高层



## 1.5.1 传统OS结构

### \* 4> 分层结构设计OS的优缺点

#### ■ 优点

- 易保证OS设计的正确性
- OS易扩充、易维护

#### ■ 缺点

- OS系统效率降低了（通信开销增加了）

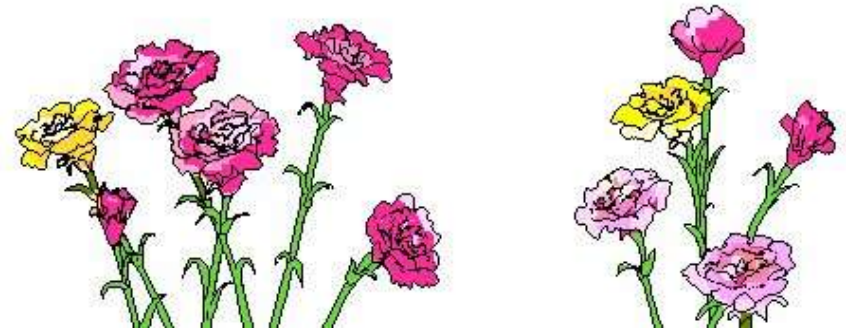




## 1.5.2 现代OS结构——微内核结构

微内核结构是20世纪80年代后期发展起来的一种OS结构，可支持多处理机运行，适用于分布式系统环境。当前比较流行的OS大多采用了微内核结构,如Mach OS、Windows 2000/XP。

- ❖ 微内核OS的基本概念
- ❖ 微内核的基本功能
- ❖ 微内核OS的优点
- ❖ 微内核OS存在的问题





## 1.5.2 现代OS结构——微内核结构

### ❖ 1、微内核OS的基本概念

#### \* 1、基于客户机/服务器模式

为了提高OS的灵活性、可扩充性、可维护性，基于客户机/服务器模式，将OS划分为两部分：

- 1> **微内核：** 仅将OS中最基本的功能、与硬件紧密相关的一些较基本的功能置于微内核中，用来处理普通用户程序（客户进程）和服务端之间的通信。
- 2> **多个服务器：** OS中的其它功能封装成若干个服务器，用来为普通用户程序（客户进程）提供服务。



## 1.5.2 现代OS结构——微内核结构

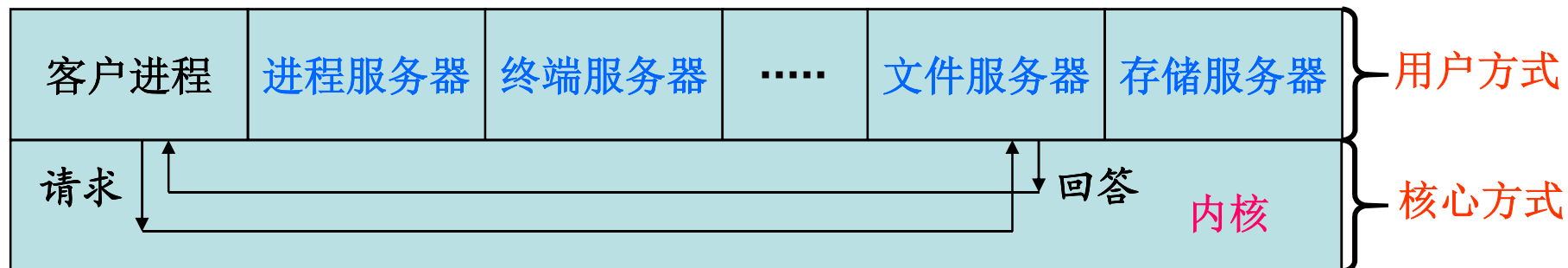


图1-7 单机环境下基于C/S模式的OS结构

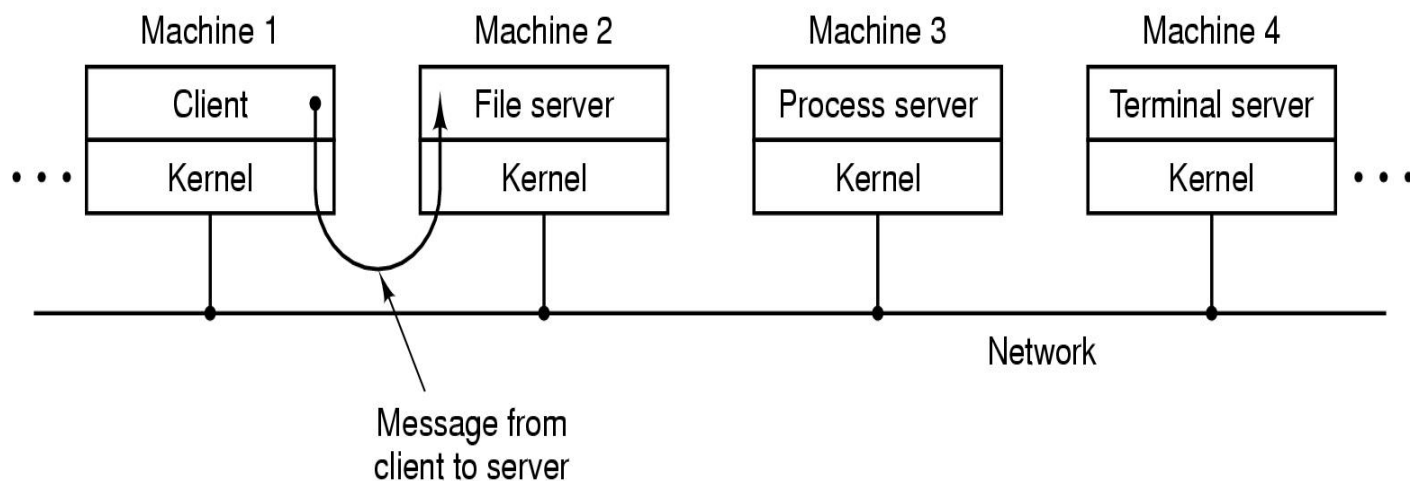


图1-8 多机环境下基于C/S模式的OS结构





## 1.5.2 现代OS结构——微内核结构

### \* 2、足够小的内核

现代OS的微内核是精心设计的、能实现OS核心功能的部分。它并非是一个完整的OS，而是OS最基本的部分。微内核更小更精练，运行于核心态（管态、系统态）、开机后常驻内存，通常用于：

- 1> 实现与硬件紧密相关的处理；
- 2> 实现一些较基本的功能；
- 3> 负责客户进程和服务端之间的通信。



## 1.5.2 现代OS结构——微内核结构

### \* 3、应用“机制与策略分离”原理

- **机制（动手干）**：是指实现某一功能的具体执行机构，在现代OS中，通常将机制置于微内核中。
- **策略（如何干）**：对机制实现的功能进行优化、扩展，在现代OS中，通常将策略置于服务器中。

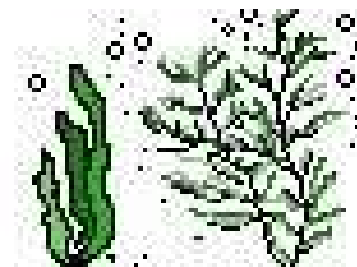




## 1.5.2 现代OS结构——微内核结构

### \* 4、采用面向对象技术

- 1> 通过重用提高OS的**质量和开发速度**
- 2> 使 OS具有更好的**可维护性和可扩展性**
  - 通过变量、方法的封装，提高OS可维护性
  - 通过类的继承，提高OS的可扩展性
- 3> 提高OS的**正确性和可靠性**
  - 类是OS的基本单元，利用单元测试可提高OS的正确性和可靠性
  - 通过封装可有效防止未经授权者对类中变量进行访问





## 1.5.2 现代OS结构——微内核结构

### ❖ 2、微内核的功能

#### \* 进程（线程）管理

- 主要是进程（线程）之间的通信管理

#### \* 低级存储器管理

- 实现存储器管理功能中与硬件相关的部分，如地址映射

#### \* 中断和陷入处理

- 中断是由CPU外部的中断源产生的，CPU被动响应
- 陷入是CPU在执行程序时由程序有意产生的，CPU主动

注：微内核OS是将进程管理、存储器管理以及I/O管理这些功能一分为二，属于机制的部分置入微内核，属于策略的部分置入服务器。即前面所讲的“应用“机制与策略分离”原理”。



## 1.5.2 现代OS结构——微内核结构

### ❖ 3、微内核OS的优点

- \* 提高了OS的可扩展性
- \* 增强了OS的可靠性
  - 服务器运行在用户态（目态），微内核运行在核心态（管态、系统态）（用户态、系统态概念详参P294）
- \* 提高了OS的可移植性
  - OS与硬件平台相关的代码在微内核中，绝大部分服务器与硬件平台无关
- \* 提供了对分布式系统的支持
  - 客户进程与服务器、服务器与服务器之间是通过消息传递通信机制进行的



## 1.5.2 现代OS结构——微内核结构

### ❖ 4、微内核OS存在的问题

较传统OS，微内核OS的运行效率有所降低，主要原因是由于客户进程与服务器之间、服务器与服务器之间的通信次数大大增加了。

为了解决这一问题，可以重新把一些常用的操作系统基本功能由服务器移入微内核中，但这也提高了微内核的设计代价。



# 本章小结

- ❖ OS的四大目标\*
  - \* OS基本定义\*\*
- ❖ OS的三大作用
- ❖ OS的五大发展动力
- ❖ OS的发展历程
  - \* OS三大基本类型\*\*
- ❖ OS的四大特性\*\*
- ❖ OS的五大功能
- ❖ OS的四种结构
  - \* 微内核结构\*\*



# 本章作业

## ❖ 要求:

- \* 大题一定要做在作业本上（**0.5-1元钱的本子，太大、太厚的拒收，力气太小搬不动哈**）

## ❖ 交作业日期:

- \* 等通知

## ❖ 作业内容:

- \* 网络测试平台第1章在线测试（等通知）
- \* 谈谈你对微内核OS结构的理解？





本章课程结束！ 谢谢大家！