

计算概论 之 认识计算机

计算机软件

刘譞哲

北京大学计算机学院



Peking
University

对软件的感性认识

如果将计算机系统比作一幢房屋，硬件可以视作钢筋水泥和框架结构，而软件则可以视作装饰、家具和电器，目的是最大程度发挥硬件的优势，给用户以舒适的使用体验



什么是计算机软件

- **应用含义**：指示计算机完成任务的，以电子格式存储的**指令序列和相关数据**
- **学科含义**：软件=程序+文档
 - 软件可以包括多个计算机程序。
 - 软件包括数据，但单独的数据不是软件
- **传统的计算机软件以产品形式（拷贝）存在**
- **版权**：授予版权所有者某种独占权利（复制、发布、出售、更改）的一种合法保护形式
- **计算机网络的发展，使得软件正在从产品转向服务，对于软件，可以“只求使用，不求拥有”**

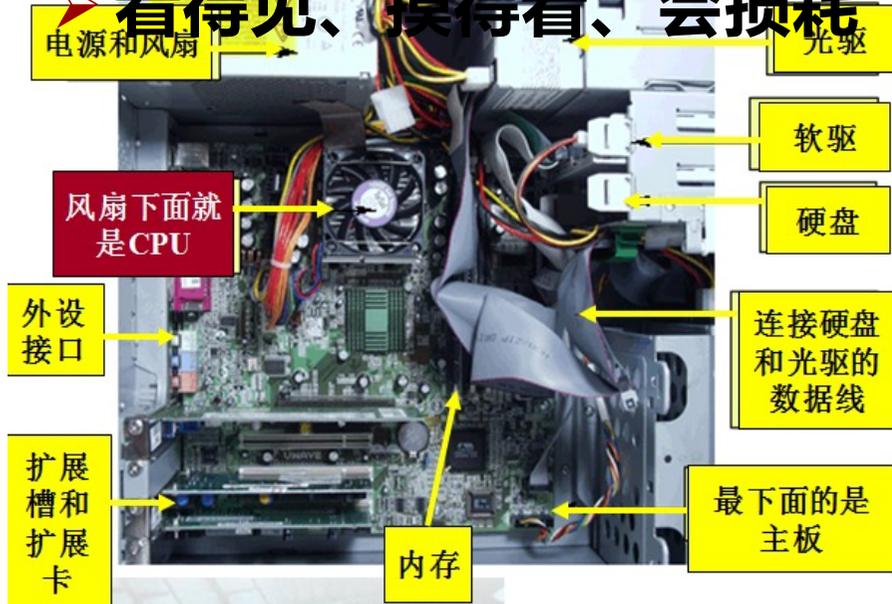


计算机硬件与软件

□ 硬件(hardware)

- 就是计算机中的所有**物理的事物**(physical artifacts)
- 包括由金属、硅、塑料等材料制成的电路、芯片、数据带、外壳、风扇等

➤ **看得见、摸得着、会损耗**

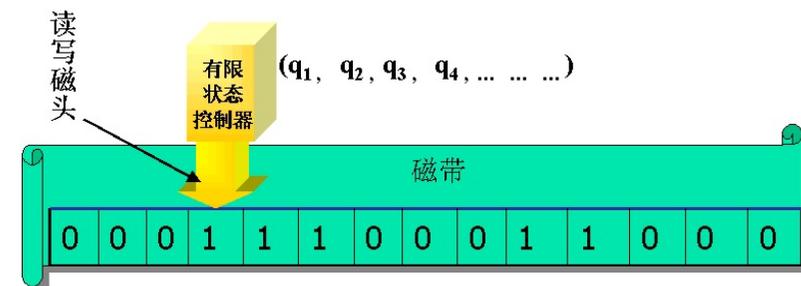


□ 软件(software)

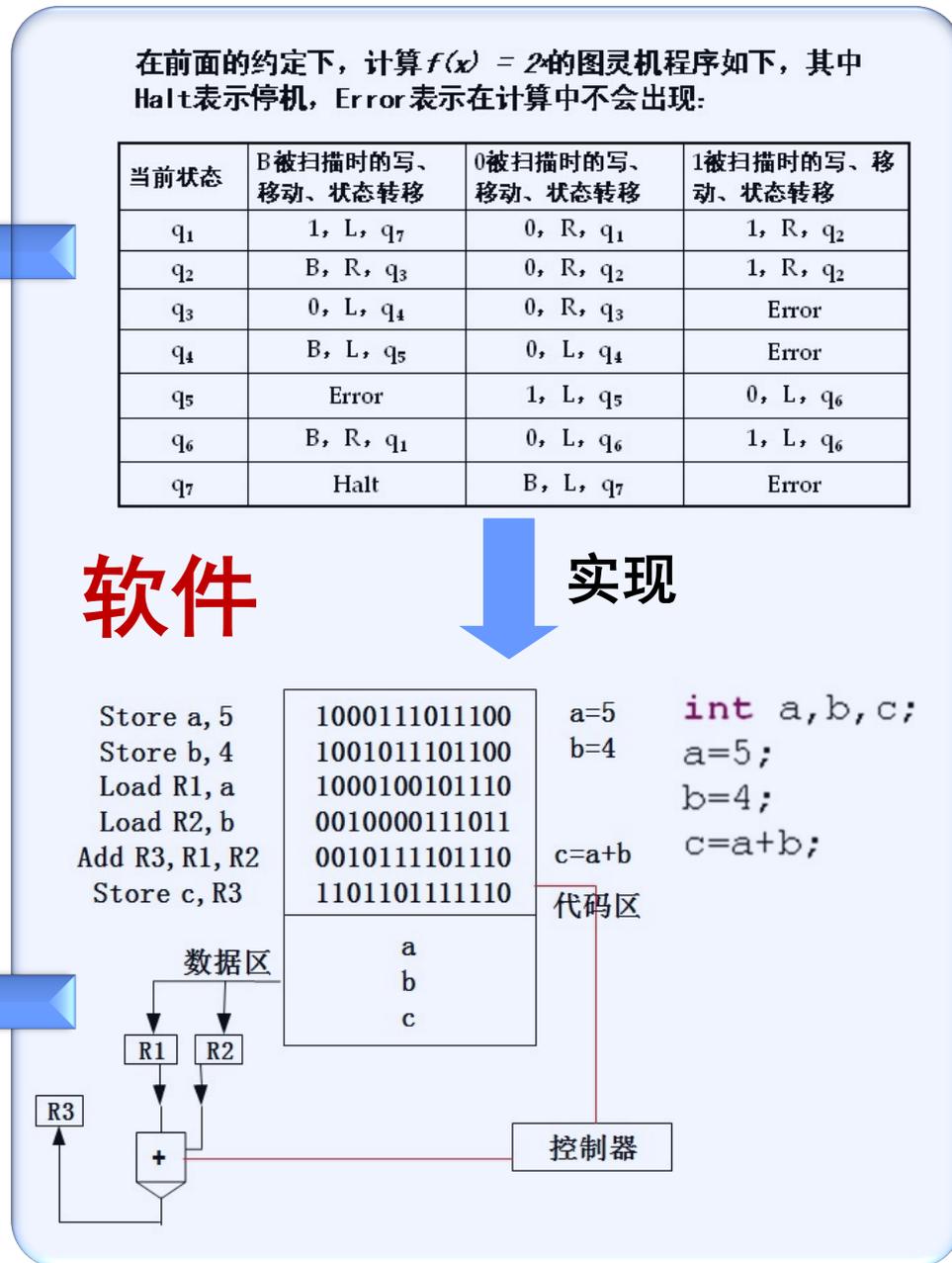
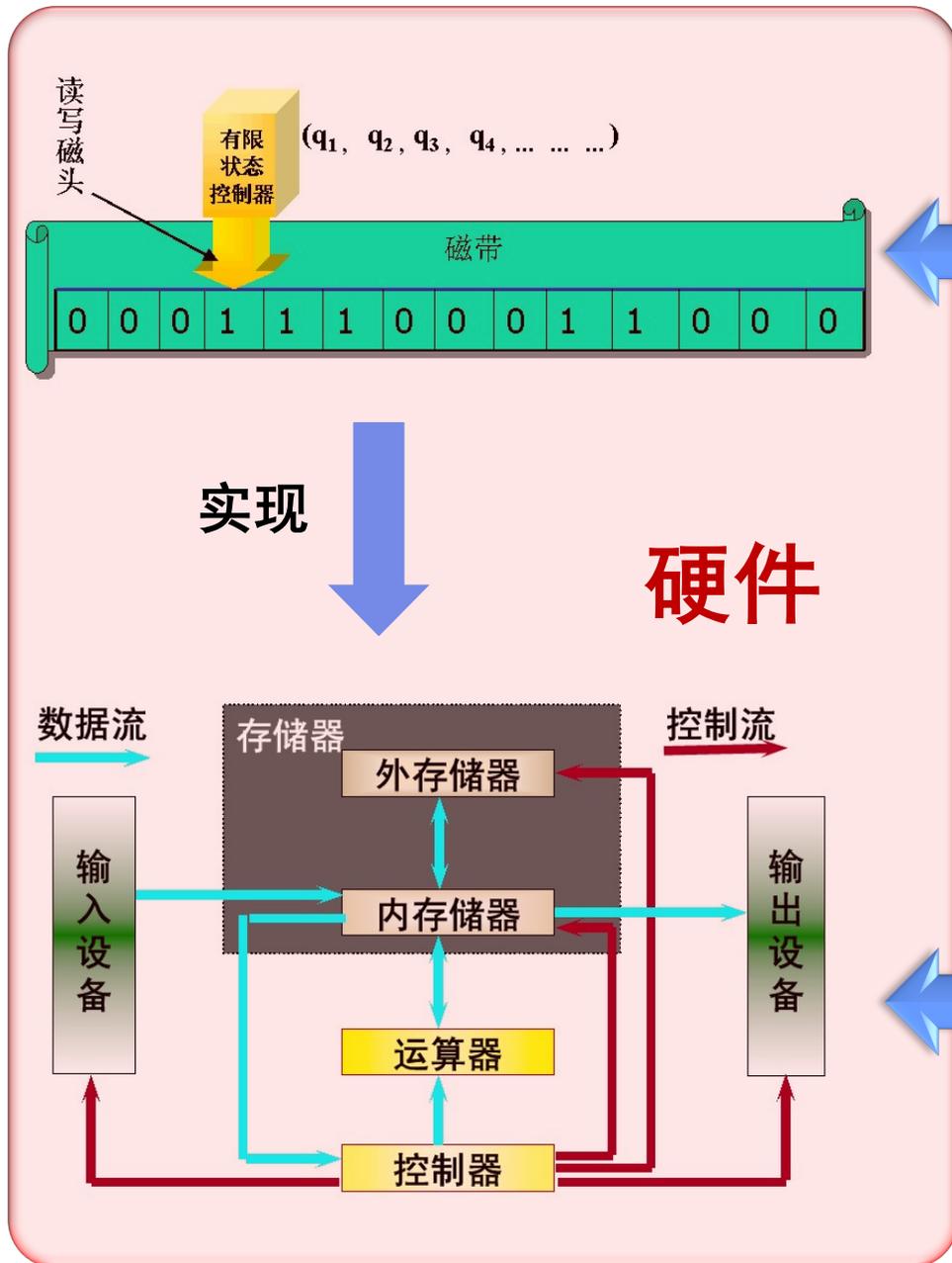
- 就是执行一次计算所需的、**硬件以外的事物** (non-physical artifacts)
- 即二进制表示的程序与数据
- 看不见、摸不着、无损耗

在前面的约定下，计算 $f(x) = 2^x$ 的图灵机程序如下，其中 Halt 表示停机，Error 表示在计算中不会出现：

当前状态	B 被扫描时的写、移动、状态转移	0 被扫描时的写、移动、状态转移	1 被扫描时的写、移动、状态转移
q ₁	1, L, q ₇	0, R, q ₁	1, R, q ₂
q ₂	B, R, q ₃	0, R, q ₂	1, R, q ₂
q ₃	0, L, q ₄	0, R, q ₃	Error
q ₄	B, L, q ₅	0, L, q ₄	Error
q ₅	Error	1, L, q ₅	0, L, q ₆
q ₆	B, R, q ₁	0, L, q ₆	1, L, q ₆
q ₇	Halt	B, L, q ₇	Error



计算机软件



计算机软件的分类

□ 计算机软件可以分为 **系统软件** 和 **应用软件**

➤ 系统软件：与应用领域无关，为其他软件提供基础设施支撑

- 操作系统：Windows，MacOS，iOS
- 程序语言环境：C语言，Basic语言，汇编语言
- 数据库：Access，IBM DB2，Oracle
- **中间件：COM**

➤ 应用类

- **各类领域、行业的应用软件**
 - 办公领域：MS Office、WPS
 - 即时通讯：QQ、MSN、Fetion
 - 各类游戏软件



计算机软件发展历经三个阶段



从“程序”发展到具有独立形态的“软件”，主要作为硬件的附属品存在；面向用户群体有限，应用领域局限（科学计算、商业计算）

以Bill Gates创立Microsoft和Larry Ellison创立Oracle为标志，软件成为独立产品，并形成巨大的产业；应用领域不断扩大（办公、游戏、嵌入式等），几乎涵盖到所有领域

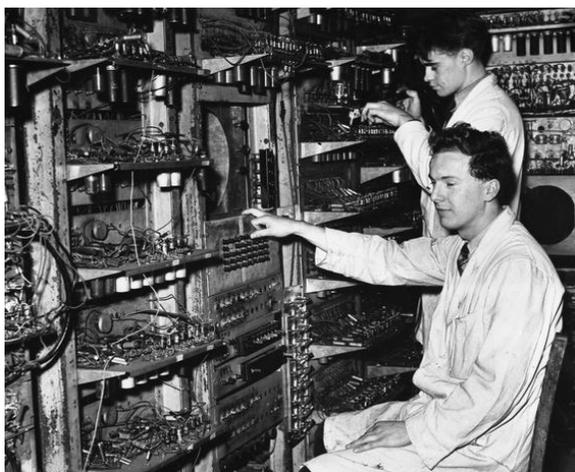
从单机环境扩展到互联网环境，开始从产品到服务，渗透到人们生活和生产的各个方面，成为互联网资源全面连接的重要使能和支撑



软硬一体化阶段 (1946 - 1975)

早期计算机没有“软件”的概念，仅仅是以程序的形式存在

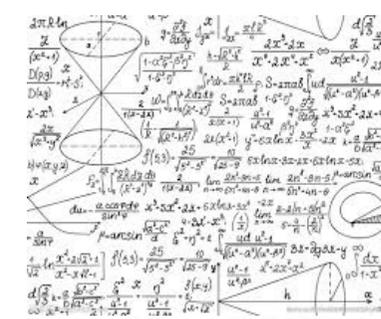
晶体管、电子管计算机



展现方式：
机器语言、汇编语言

```
0000000 0000 0001 0001 1010 0010 0001 0004 0128
0000010 0000 0016 0000 0028 0000 0010 0000 0020
0000020 0000 0001 0004 0000 0000 0000 0000 0000
0000030 0000 0000 0000 0010 0000 0000 0000 0204
0000040 0004 0004 0004 c7c8 00c8 4748 0048 e8a9
0000050 00a9 6a69 0069 a0a9 00a9 2028 0028 fdfc
0000060 00fc 1819 0019 0098 0098 d9d8 00d8 5057
0000070 0057 7b7a 007a bab9 00b9 3a3c 003c 8888
0000080 8888 8888 8888 8888 200e ba88 8888 8888
0000090 3b83 5788 8888 8888 7667 778e 8828 8888
00000a0 d61f 7-bd 8818 8888 467c 505f 8814 8188
00000b0 8b06 e0f7 88aa 8388 8b3b 80f9 88bd e988
00000c0 8a18 880c e841 c988 b328 6871 688e 958b
00000d0 a948 5062 5884 7e81 3788 1ab4 5a84 3e8c
00000e0 3d86 dcb8 5cbb 8888 8888 8888 8888 8888
00000f0 8888 8888 8888 8888 8888 8888 8888 0000
0000100 0000 0000 0000 0000 0000 0000 0000 0000
*
0000130 0000 0000 0000 0000 0000 0000 0000
000013e
```

应用领域：
军事领域的计算为主



成功运行，可视为“软件”的诞生



Peking University

产品化、产业化阶段 (1975 -)

- 以Microsoft和Oracle出现，标志着软件开始成为一个独立产业
- PC的广泛应用和软件产品化催生了信息化的第一波浪潮，即以单机应用为特征的数字化阶段（信息化1.0）

个人计算机的广泛应用 以拷贝为主要形态



以IBM PC为代表

ORACLE

Larry Ellison
创建Oracle，
成为第一个
纯“软件公
司”



Microsoft



WinTel联盟的出现
彻底改变了整个信
息产业的格局

应用领域：

几乎扩展到了所有应用领域



桌面操作系统



嵌入式系统



办公软件



企业资源规划



Peking
University

网络化、服务化阶段 (1995 -)

互联网推动了软件从单机向网络计算环境的延伸，带来了信息化的第二波浪潮，进入以联网应用为特征的网络化阶段（信息化2.0）

互联网作为平台

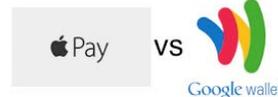


形态：
传统拷贝+服务+Apps



应用领域：

社会经济生活的方方面面

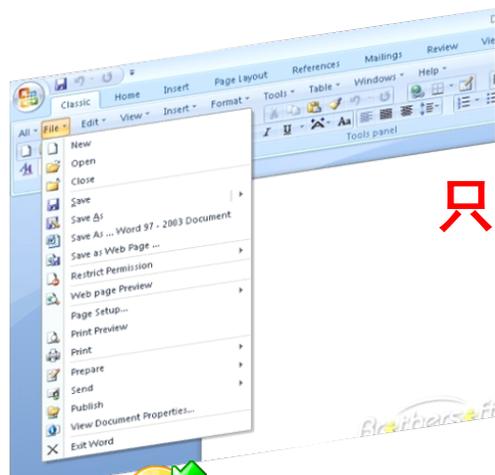


零售业、批发业、制造业、机器人、汽车、能源、广告业、新闻业、通信业、物流业、酒店业与旅游行业、餐饮业、金融业、保险业、医疗业、教育行业、电视节目行业、电影行业、出版业……

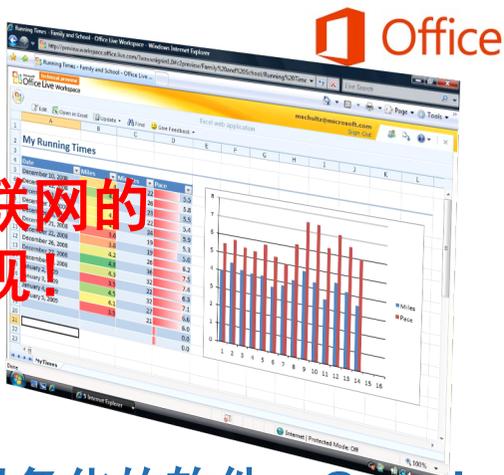


Peking University

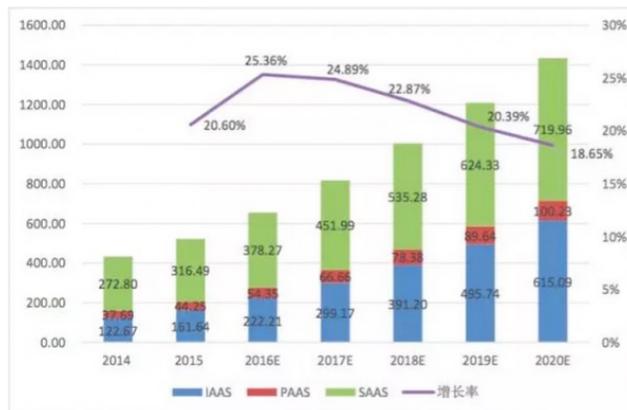
互联网环境下软件使用新方式--服务化



只有软件和互联网的
结合才能实现!



Office



单机环境下的软件：
Office本地拷贝+
安装+升级

服务化的软件：Google Docs/Office 365, 在线直接使用+永远的“beta”版

Gartner报告称2016年全球公共云服务市场规模达到2040亿美元，其中SaaS为277亿美元，年增长率为20%

典型服务化应用案例



Peking University

互联网环境下软件新拷贝模式--App

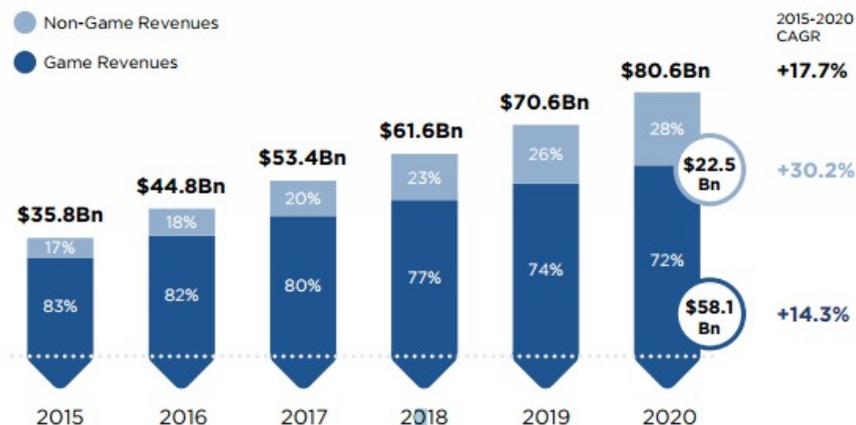


Apple AppStore 和 Google Play 拥有**300万+**个 App 累计下载量均超过**1500亿次**

应用商店是互联网时代软件拷贝的新型分发和盈利模式

GLOBAL APP STORE REVENUES

2015-2020 | GAME & NON-GAME REVENUE SPLIT | 2015-2020 CAGR



2016 年 全球移动商店收入约为**448亿美元**
 iOS 应用商店和 Google Play 的收入超过 **350 亿**
 美元，年增长率高达 **40%**



Peking University

为什么要操作系统?

输入设备

主机箱

如何管理种类繁多的外设?

如何同时处理多个任务?

设备管理

进程管理

内存管理

文件管理

操作系统
Operating System

用户界面

方便地...
主机?

输出

输入

如何...
的数据交换?

磁带

外设

Windows Mess...
PKIAS Features in IonAS 5 (联机)
CPU (2/4)
所有联系人 (9/16)

计算器

正在播放

计算概论

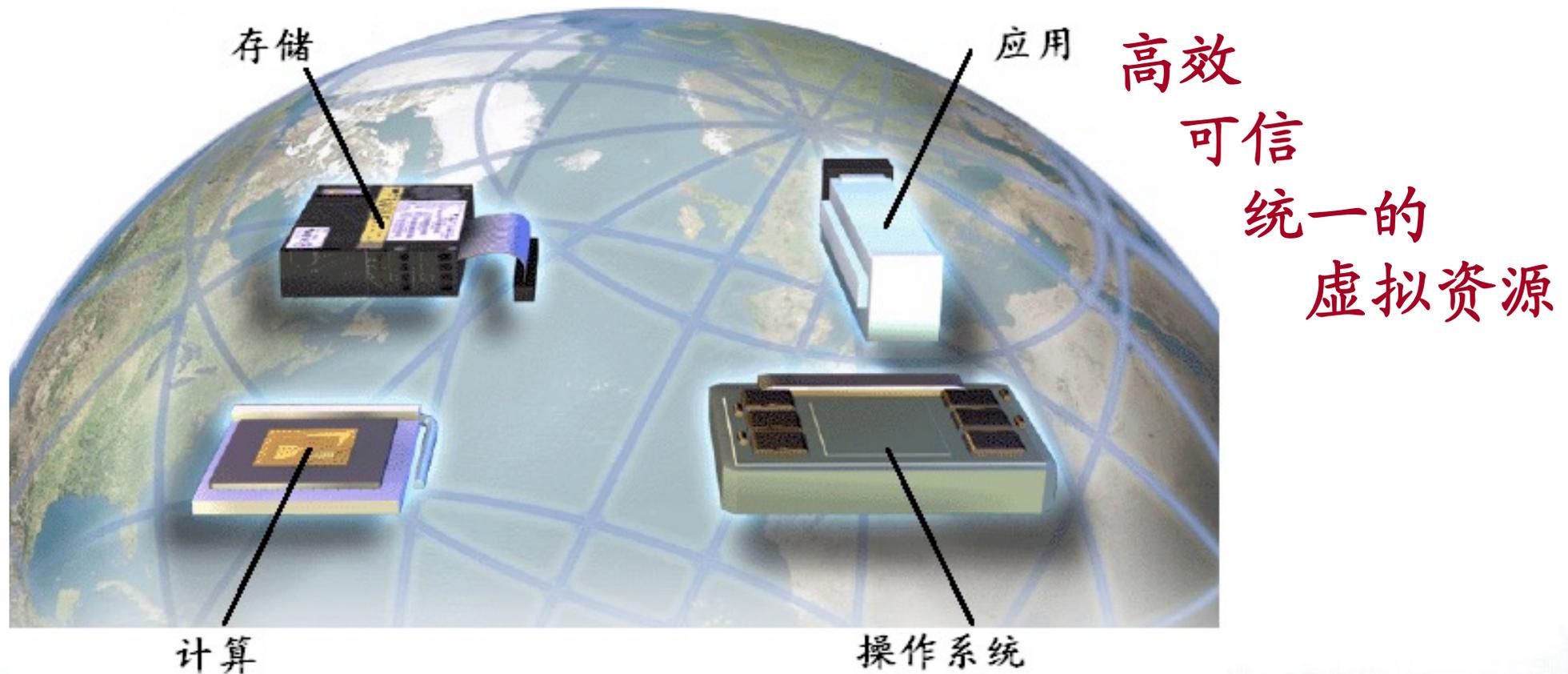
Google 桌面 - Microsoft Internet Explorer

未命名 - 画图

如何...
的数据交换?

操作系统的定义

- **操作系统**是计算机系统中的一个系统软件，是一些**程序模块的集合**，以追求更高效、合理地**发挥硬件**所提供的**计算能力**



一些著名的操作系统

□ 工作站、大型机操作系统

➤ **UNIX**

□ 个人计算机操作系统

➤ **DOS**

➤ **Windows**

➤ **Macintosh OS**

➤ **Linux**



UNIX

□ 现代操作系统的代表

- 用**C**语言编写，因此它是可移植的
 - **UNIX**是世界上唯一能在笔记本电脑、**PC**机、工作站直至巨型机上运行的操作系统
- 是一个良好的、通用的、多用户、多任务、分时操作系统
- Ken Thompson, Dennis Ritchie
 - 1983年图灵奖
- 研发动机之一：可以在DEC PDP-7小型计算机上玩星际探险游戏



DOS

□ Disk Operation System

- 个人计算机的成功，逼得**IBM**急需现成的操作系统
- **IBM**公司洽谈**CP/M**操作系统不顺利，机遇落到了微软公司
- 微软经销西雅图计算机产品公司的**QDOS**操作系统
- 如果当时西雅图公司知道**QDOS**将被转卖给**IBM**，历史将会怎样？
- **IBM PC**和**MS DOS**于**1981**年推出
 - **Microsoft**的第一桶金
- **1995**年被**Windows**正式取代
 - 最大的困难是使用不方便(命令行界面)
 - 单用户，只能使用**640K**内存



DOS

```
c:\ E:\WINDOWS\system32\cmd.exe
E:\>cd ejbtest

E:\EJBTest>dir/w
驱动器 E 中的卷没有标签。
卷的序列号是 D0A6-10F9

E:\EJBTest 的目录

[.]                [...]
[bak]              [classes]
[doc]              EJBTest.html
EJBTest.html~    EJBTest.html~1~
EJBTest.jpj       EJBTest.jpj.local
EJBTest.jpj.local~ EJBTest.jpj~
HelloSessionBean.ejbgrpj MyBean.ejbgrpj
MyBean.jar        [src]
[TestEJBClient]

                10 个文件          11,225 字节
                7 个目录    1,663,692,800 可用字节

E:\EJBTest>list
'list' 不是内部或外部命令，也不是可运行的程序
或批处理文件。

E:\EJBTest>
```



Windows

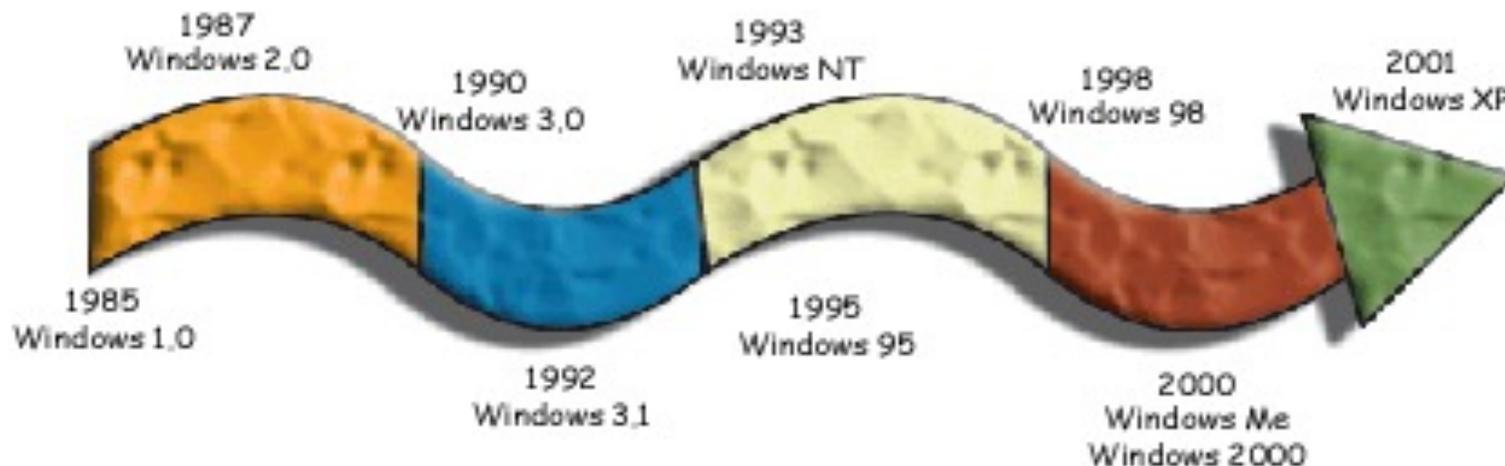
□ 微软成为全球软件巨头的依赖

➤ Windows 3.1发布

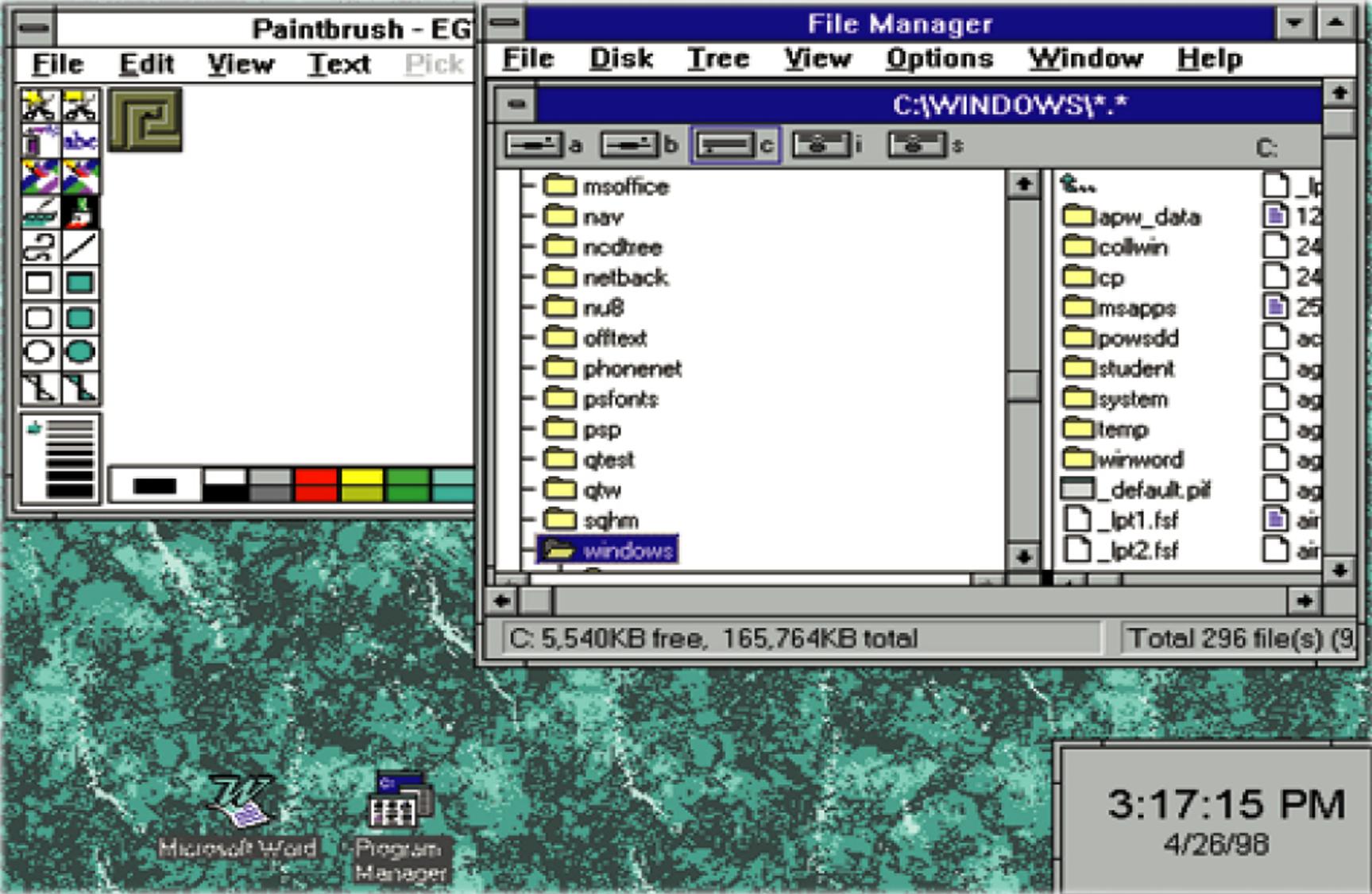
- 一个划时代的产品
- 运行于**DOS**之上
- 还不是真正的操作系统，只是“窗口”

➤ Windows95较为完整的操作系统

“傻瓜”
“好用”
最大的特点就是



Windows



Windows 3.1



Peking University

Mac OS

□ 苹果Macintosh系列电脑上的操作系统

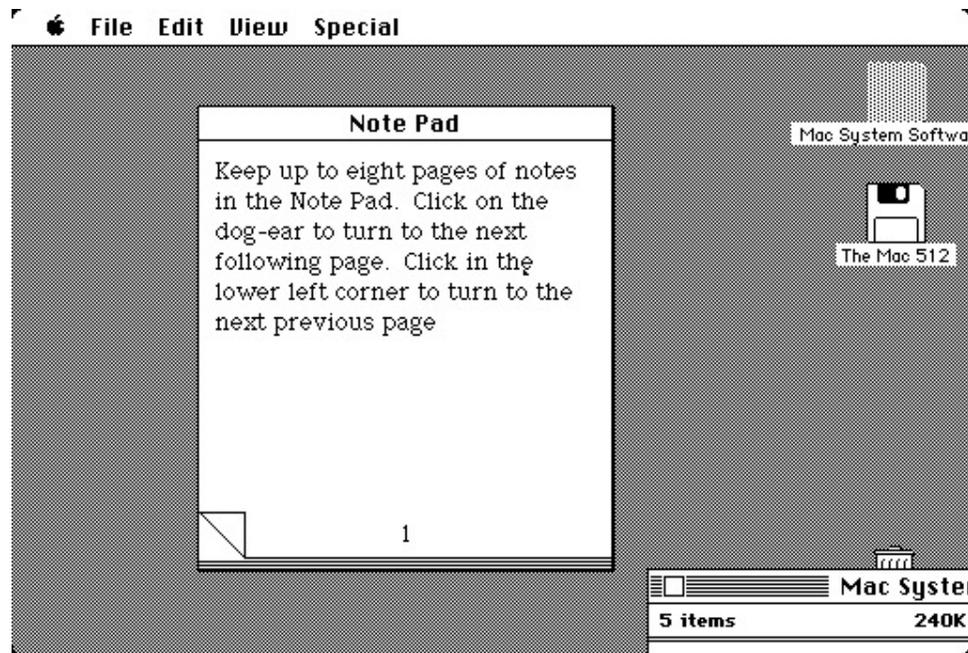
- 首个在商用领域成功的图形用户界面
- **1984**年第一版
- 在图形图像处理占垄断地位



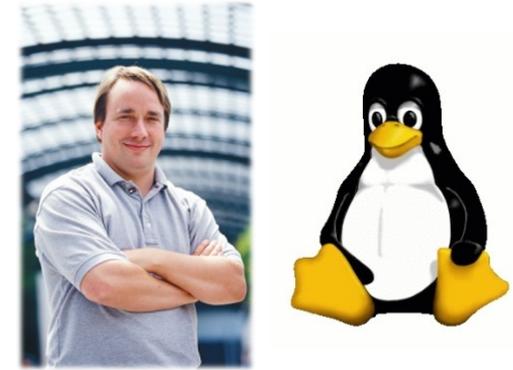
□ 源自施乐Palo Alto研究中心

- **70**年代的计算机研究思想库
- 世界上第一台个人计算机**Alto**于**1972**年在这里出现
- 图形界面、手持鼠标、面向对象程序设计、微机网络、桌面出版和激光打印、面向侧面的编程等等具有先进概念和技术的原型都首次出现在这里

Mac OS



Linux

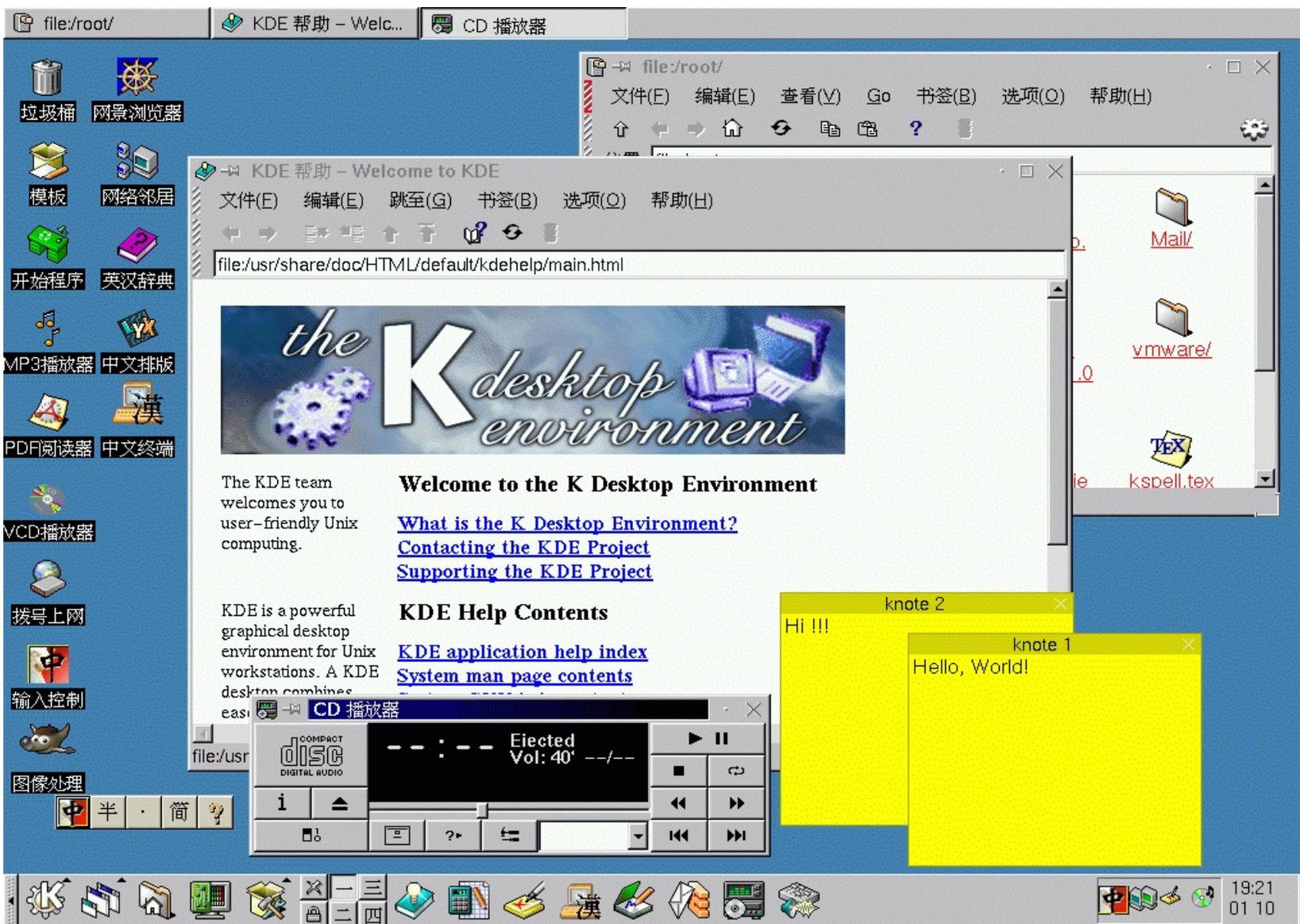


□著名的微机开源操作系统

➤其创始人是**Linus Torvalds**

- **Linus**需要终端仿真程序来存取**Usenet**新闻组的内容，于是他编写了从调制解调器上接发信息的程序以及显示器、键盘和调制解调器的驱动程序
- 然后编写了磁盘驱动程序、文件系统，一旦有了进程切换、文件系统和设备驱动程序，当然就拥有了一个操作系统原型，或者至少是它的一个内核
- 类**Unix**操作系统，目前主要用于构造各种服务器端应用，成为**Windows**的主要竞争对手
- 开源：在遵循相应规范以及知识产权规定的前提下，每个人可以参与其开发

Linux



智能手持设备操作系统



Apple iOS



Android OS



Web OS



Windows Mobile OS

智能手持设备操作系统和桌面操作系统比较

- 都提供了资源管理、设备管理、用户界面等基本功能
- 智能手持设备操作系统可以直接放在**ROM**中，而不需要从硬盘加载到**RAM**
- 智能手持设备计算能力和存储能力正在不断增强

智能手持设备的操作系统



苹果iOS是由苹果公司开发的手持设备操作系统。最早于2007年1月9日的Macworld大会上公布。最初是设计给iPhone使用的，后来陆续套用到iPod touch、iPad以及Apple TV等苹果产品上

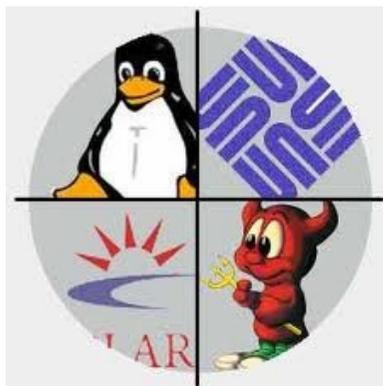
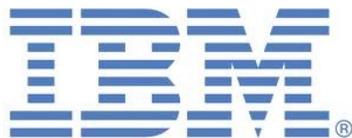
Android是一种以Linux为基础的开放源代码操作系统。2003年投入研发，2005年被Google收购。由于其开放平台的特性，被多家手机制造厂商如三星、摩托罗拉、LG采用。



□ iOS和Android推动了互联网环境下移动设备应用（即Apps）的发展和繁荣，改变了软件发布模式



操作系统发展—面向单个计算设备



1956年：
第一个实用的
OS (IBM 704)：
GM-NAA I/O

1960s：
大型机/主机OS
IBM OS/360系列

1970s：
小型机/工作站OS
UNIX

1980-90s: PC OS
Mac, Windows,
Linux

2000s: 智能手机OS
iOS, Android, WP7
MeeGo

- 单通道批处理
- I/O管理

- 批处理/分时系统
- 设备管理
- 内存管理
- 虚拟机

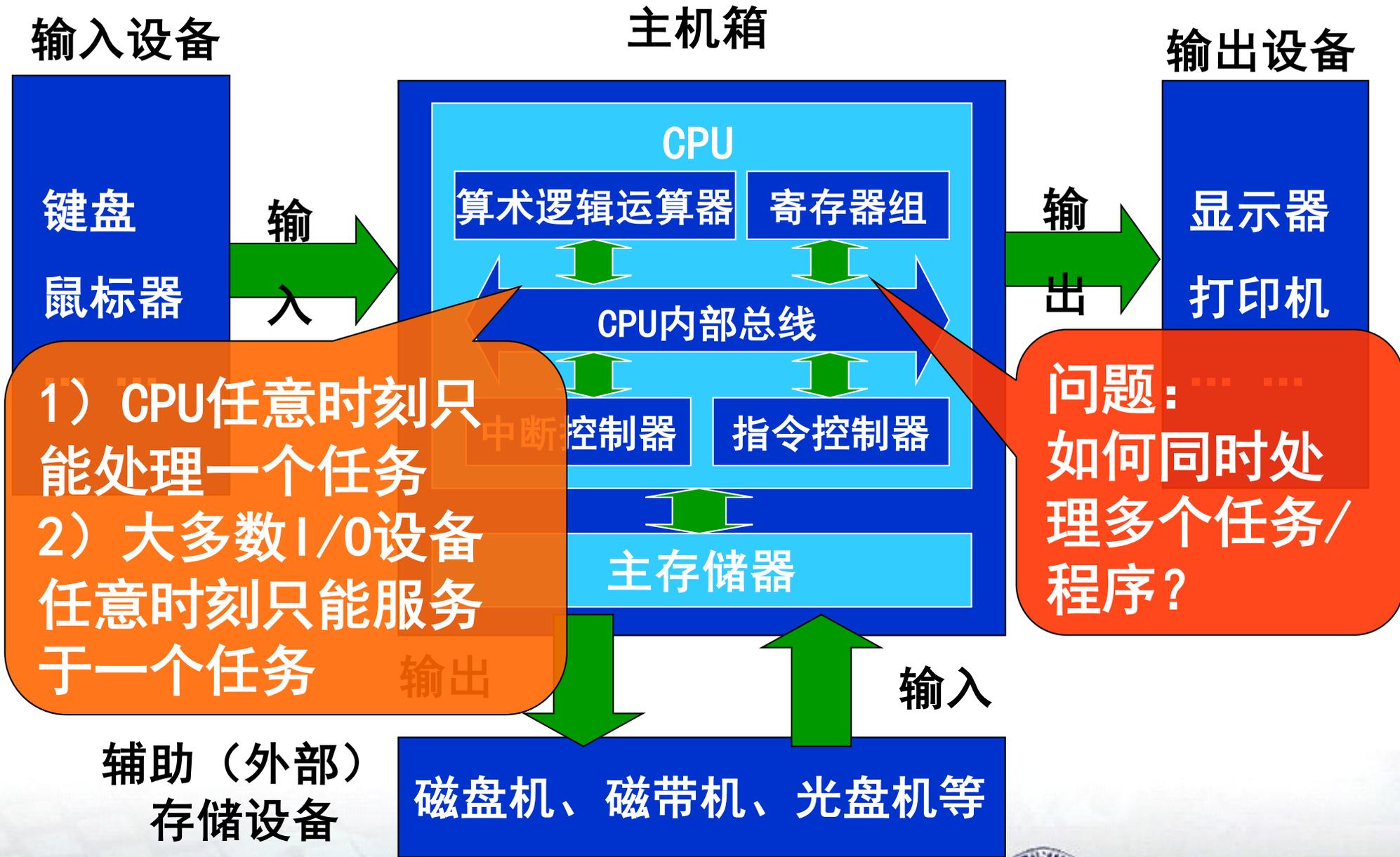
- 第一个现代OS
- 使用机器无关语言 (C语言) 开发
- 提供了标准化接口
- 集成了开发环境

- 提供现代图形用户界面 (GUI)

- 传统操作系统的裁剪
- 着眼于易用性
- 新型应用软件发布模式 (App Store、Android Market)



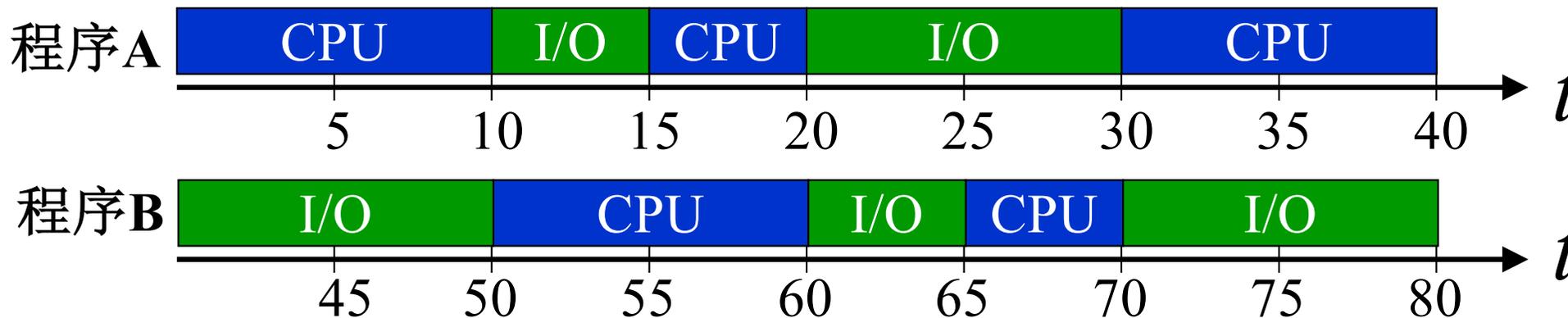
为什么需要进程管理



1) CPU任意时刻只能处理一个任务
2) 大多数I/O设备任意时刻只能服务于一个任务

问题：……
如何同时处理多个任务/程序？

为什么需要进程管理

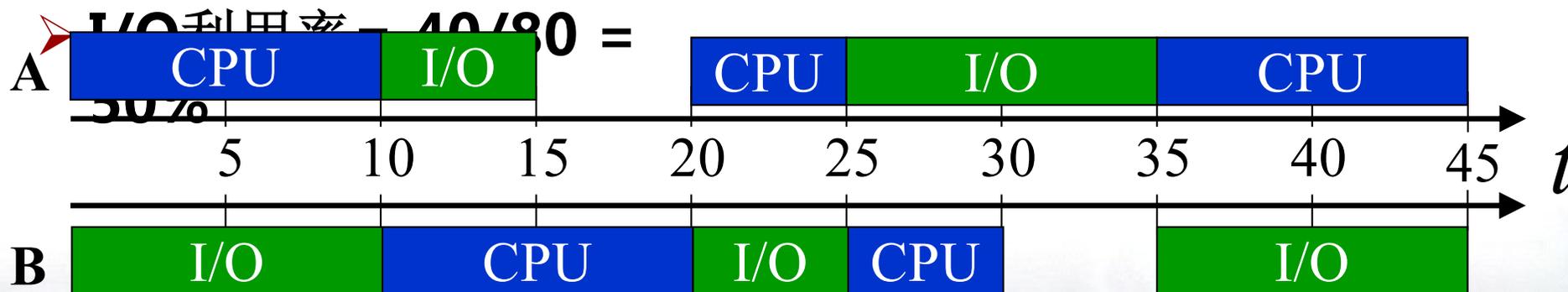


程序的顺序运行

- A先运行，B再运行
- CPU利用率 = $40/80 = 50\%$

程序的并发运行

- A、B同时运行
- CPU利用率 = $40/45 = 89\%$
- I/O利用率 = $40/45 = 89\%$



为什么需要进程管理

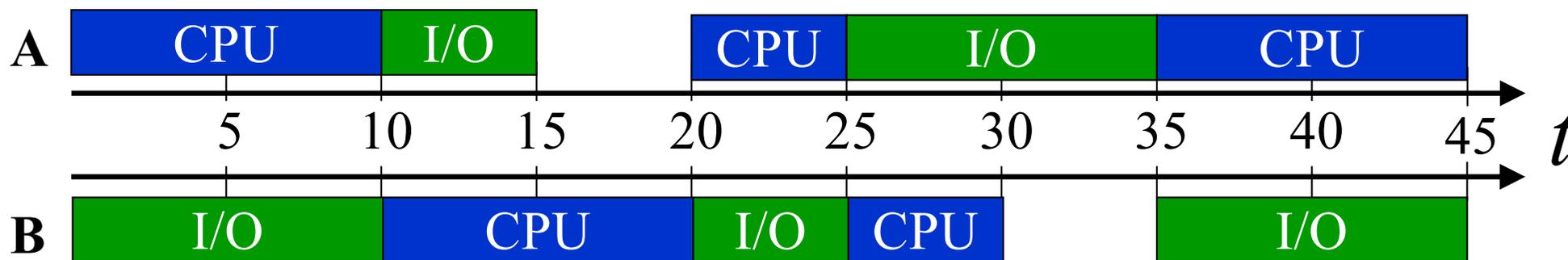
□ 能否直接针对程序进行CPU调度和资源分配？

➤ NO

➤ 因为程序主要描述完成计算的步骤，但一般

- 不考虑多个程序的并发
- 某个程序可能需要多个任务并发执行

➤ 所以，需要一种新的概念/机制——进程



进程：使能程序并发运行

□ Process（又称为任务，task）

- 是具有独立功能的程序在某个数据集合上进行的一次运行活动
- 是系统进行资源分配和调度的独立单位

□ 进程映像

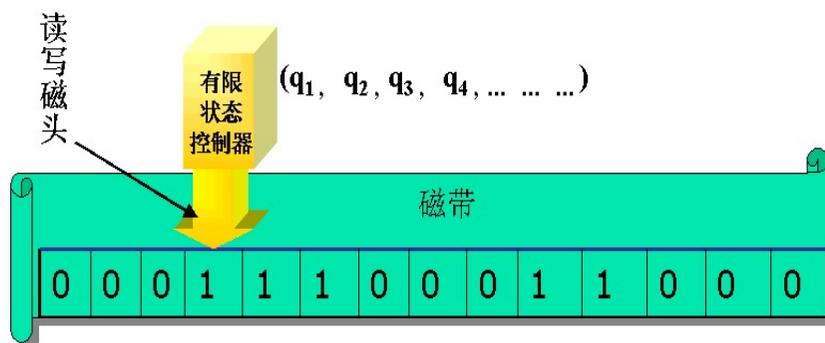
- 用户程序（代码）
- 用户数据
- 堆栈（用于过程调用和参数传递）
- 进程控制块**PCB** (进程属性)



程序与进程

□ 程序与进程的区别

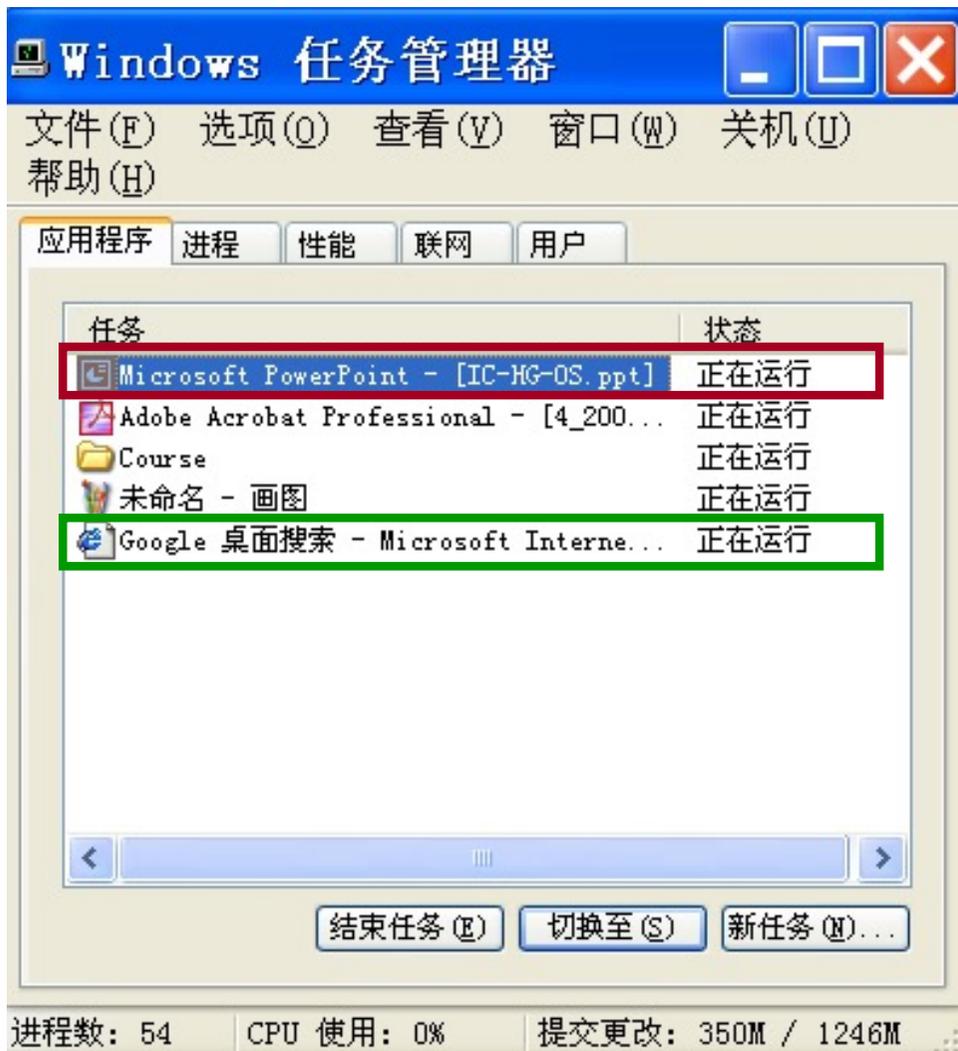
- 进程更能真实地描述并发，而程序不能
- 进程是由程序和**数据**两部分组成的
- 程序是静态的，进程是**动态的**，描述程序执行时动态特征
- 进程有生命周期，有诞生有消亡，短暂的；而程序是相对长久的
- 一个程序可对应多个进程，反之亦然
- 进程具有创建其他进程的功能，而程序没有



在前面的约定下，计算 $f(x) = 2^x$ 的图灵机程序如下，其中 Halt 表示停机，Error 表示在计算中不会出现：

当前状态	B被扫描时的写、移动、状态转移	0被扫描时的写、移动、状态转移	1被扫描时的写、移动、状态转移
q ₁	1, L, q ₇	0, R, q ₁	1, R, q ₂
q ₂	B, R, q ₃	0, R, q ₂	1, R, q ₂
q ₃	0, L, q ₄	0, R, q ₃	Error
q ₄	B, L, q ₅	0, L, q ₄	Error
q ₅	Error	1, L, q ₅	0, L, q ₆
q ₆	B, R, q ₁	0, L, q ₆	1, L, q ₆
q ₇	Halt	B, L, q ₇	Error

程序与进程



Windows 任务管理器

文件(F) 选项(O) 查看(V) 窗口(W) 关机(U)
帮助(H)

应用程序 进程 性能 联网 用户

任务	状态
Microsoft PowerPoint - [IC-MG-OS.ppt]	正在运行
Adobe Acrobat Professional - [4_200...	正在运行
Course	正在运行
未命名 - 画图	正在运行
Google 桌面搜索 - Microsoft Interne...	正在运行

结束任务(E) 切换至(S) 新任务(N)...

进程数: 54 CPU 使用: 0% 提交更改: 350M / 1246M



Windows 任务管理器

文件(F) 选项(O) 查看(V) 关机(U) 帮助(H)

应用程序 进程 性能 联网 用户

映像名称	用户名	CPU	内存使用
DrvMon.exe	Owner	00	3,912 K
explorer.exe	Owner	00	4,748 K
GoogleDesktop.exe	Owner	00	9,108 K
GoogleDesktopCrawl.exe	Owner	00	4,304 K
GoogleDesktopIndex.exe	Owner	00	7,232 K
GoogleDesktopOE.exe	Owner	00	1,592 K
GoogleDesktopQQPlug...	Owner	00	3,800 K
lsass.exe	SYSTEM	00	876 K
mspaint.exe	Owner	00	1,396 K
wowexec.exe	Owner	00	...
study.exe	Owner	00	1,872 K
POWERPNT.EXE	Owner	00	1,496 K
RegSvc.exe	SYSTEM	00	3,356 K
Rtvscan.exe	SYSTEM	00	22,964 K
rundll32.exe	Owner	00	3,876 K
S24EvMon.exe	SYSTEM	00	2,612 K
SavRoam.exe	SYSTEM	00	4,152 K
scardsvr.exe	LOCAL SERVICE	00	2,720 K

显示所有用户的进程(S) 结束进程(E)

进程数: 52 CPU 使用: 3% 提交更改: 329M / 1246M

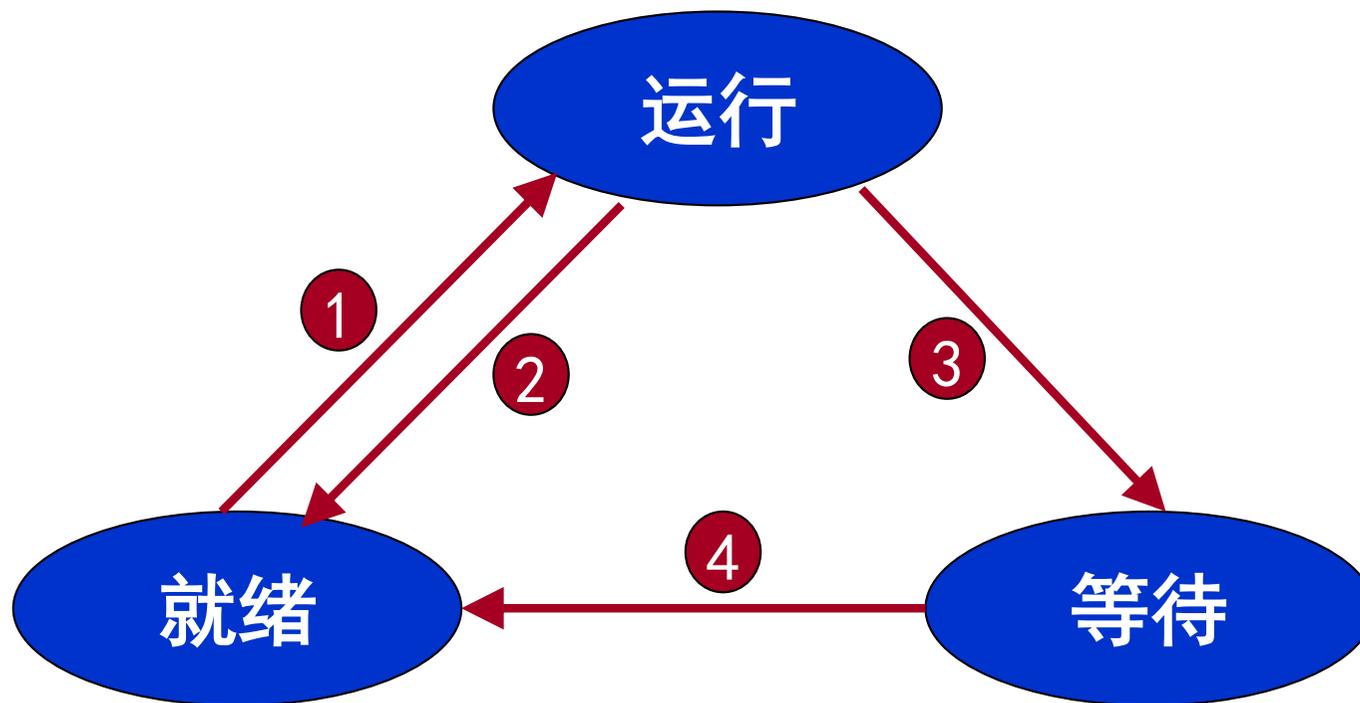
进程的分类：系统进程和用户进程（前者优于后者）



进程的状态

□ 进程的三种基本状态

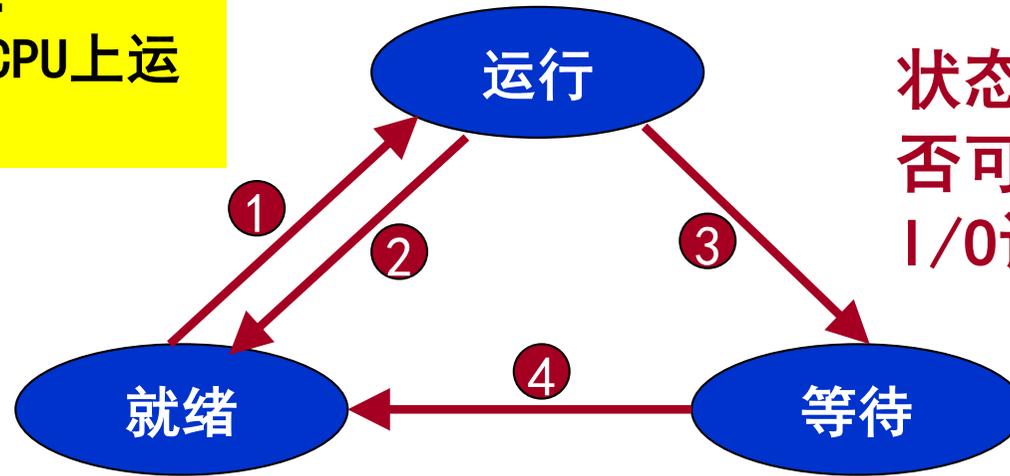
- 进程在生命消亡前处于且仅处于三种基本状态之一
- 表明进程是否可以使用**CPU**和**I/O**设备
- 不同系统设置的进程状态数目不同（5，7）



进程标识
进程状态
进程控制信息
堆栈
程序+数据
共享地址空间

进程的状态

运行态 (Running) :
进程占有CPU, 并在CPU上运行



状态表明进程是否可以使用CPU和I/O设备

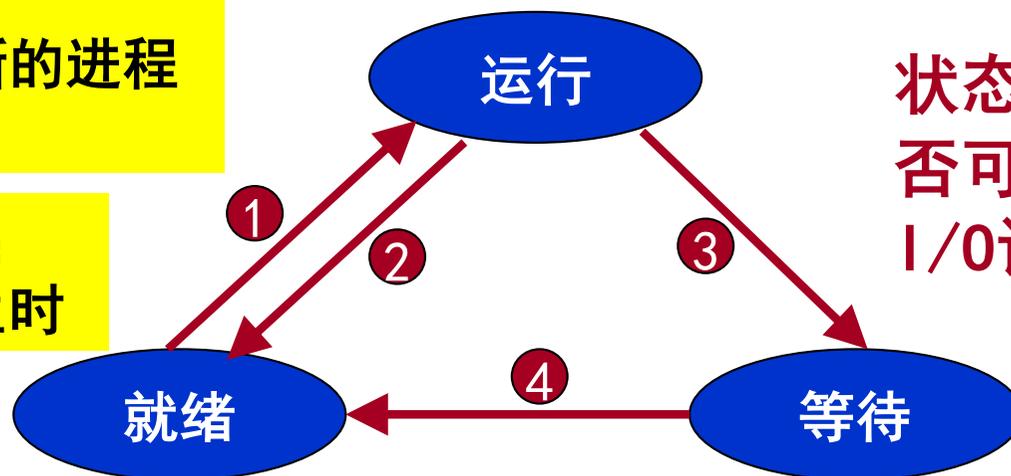
就绪态 (Ready) :
一个进程已经具备运行条件, 但由于无CPU暂时不能运行的状态 (当调度给CPU时, 立即可以运行)

等待态 (Blocked) :
阻塞态、封锁态、睡眠态
指进程因等待某种事件的发生而暂时不能运行的状态 (即使CPU空闲, 该进程也不可运行)

进程的状态转换

(1) 就绪→ 运行:
调度程序选择一个新的进程运行

(4) 等待→ 就绪:
当所等待的事件发生时



状态表明进程是否可以使用CPU和I/O设备

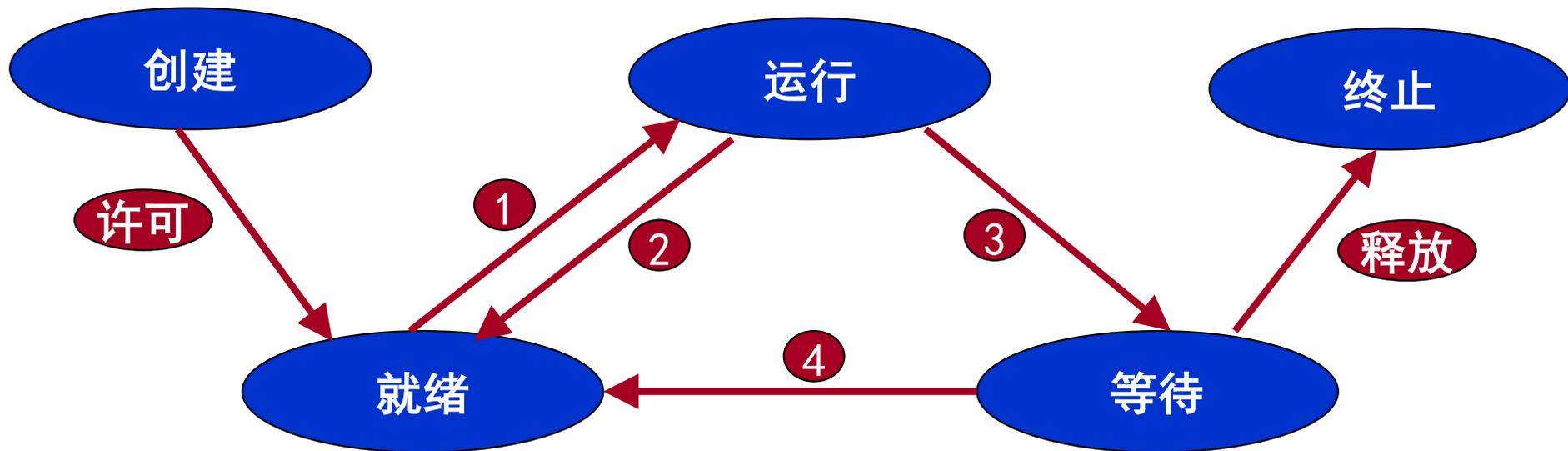
(2) 运行→ 就绪:
运行进程用完了时间片, 或者运行进程被中断, 因为一高优先级进程处于就绪状态

(3) 运行→ 等待:
当一进程必须等待时
→ OS尚未完成服务
→ 对一资源的访问尚不能进行
→ 初始化I/O 且必须等待结果
→ 等待某一进程提供输入

- 在进程运行过程中, 由于进程自身进展情况及外界环境的变化, 这三种基本状态可以依据一定的条件相互转换
- 状态的转换历史就是进程对**CPU**和**I/O**设备的使用历史



进程的其他状态



□ 创建状态

- 操作系统已完成为创建一进程所必要的工作
- 但还没有允许执行该进程，因为资源有限

□ 终止状态

- 中止后进程移入该状态，它不再有执行资格

进程控制块

□ Process Control Block, PCB

- 系统为了管理进程设置的一个专门的数据结构
- 进程与**PCB**是一一对应的

进程描述信息

进程控制信息

CPU现场保
护信息

所拥有的资源和
使用情况

Process management

Registers
Program counter
Program status word
Stack pointer
Process state
Priority
Scheduling parameters
Process ID
Parent process
Process group
Signals
Time when process started
CPU time used
Children's CPU time
Time of next alarm

Memory management

Pointer to text segment
Pointer to data segment
Pointer to stack segment

线程：
轻量级进程

File management

Root directory
Working directory
File descriptors
User ID
Group ID

进程标识

进程状态

进程控制信息

堆栈

程序+数据

共享地址空间



进程与线程

□ Thread

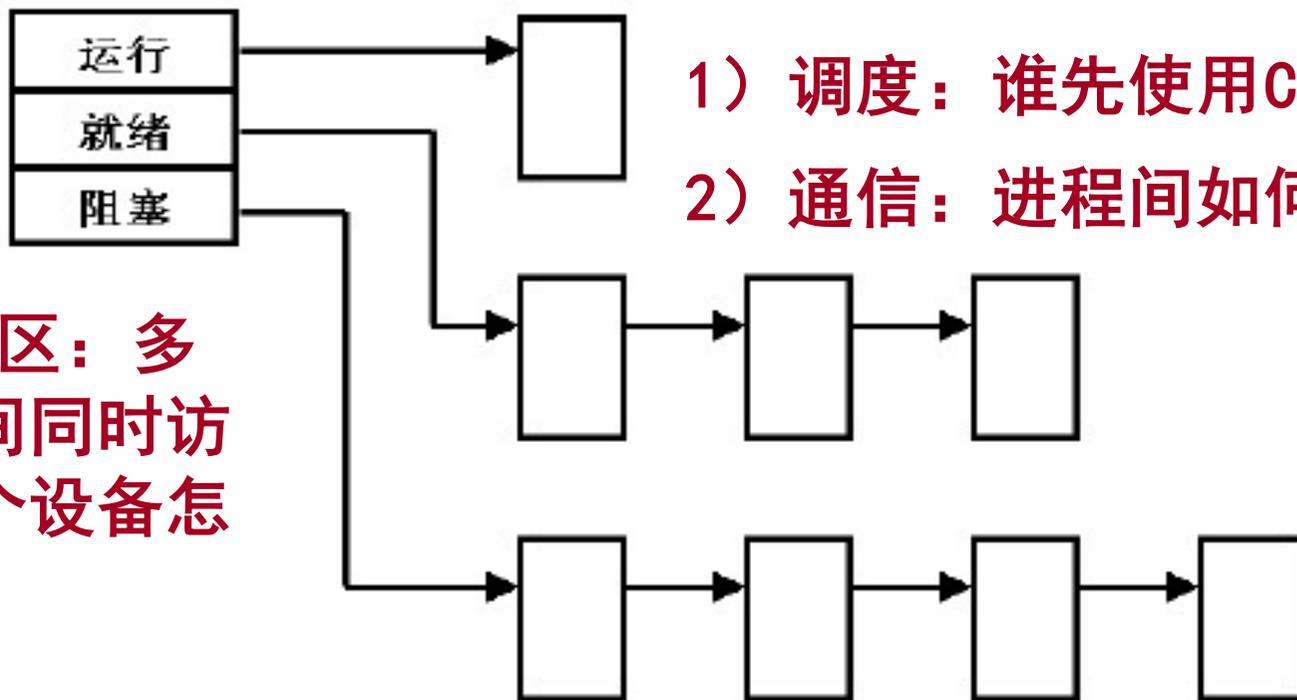
- 是进程中的一个实体，是被**CPU**执行任务的实体
- 线程自己不拥有系统资源，只拥有在执行中必不可少的资源(如程序计数器、一组寄存器和栈等)，但是它可以和同属一个进程的其他线程共享进程所拥有的全部资源
- 更适合于多任务调度
 - 便于将进程分割成多个并发单元
 - 线程附带的资源少因而创建更快



进程管理的其他问题

□ PCB表

- 系统把所有**PCB**组织在一起，并把它们放在内存的固定区域
- 其大小决定了系统中最多可同时存在的进程个数，称为系统的并发度



1) 调度：谁先使用CPU?

2) 通信：进程间如何交换数据?

3) 临界区：多个进程间同时访问同一个设备怎么办?

4) 如何避免多个进程之间产生死锁?

操作系统的功能



进程管理：
调度CPU和分配系统资源

为什么要存储管理

□ 帕金森定律：“内存多大，程序多长”

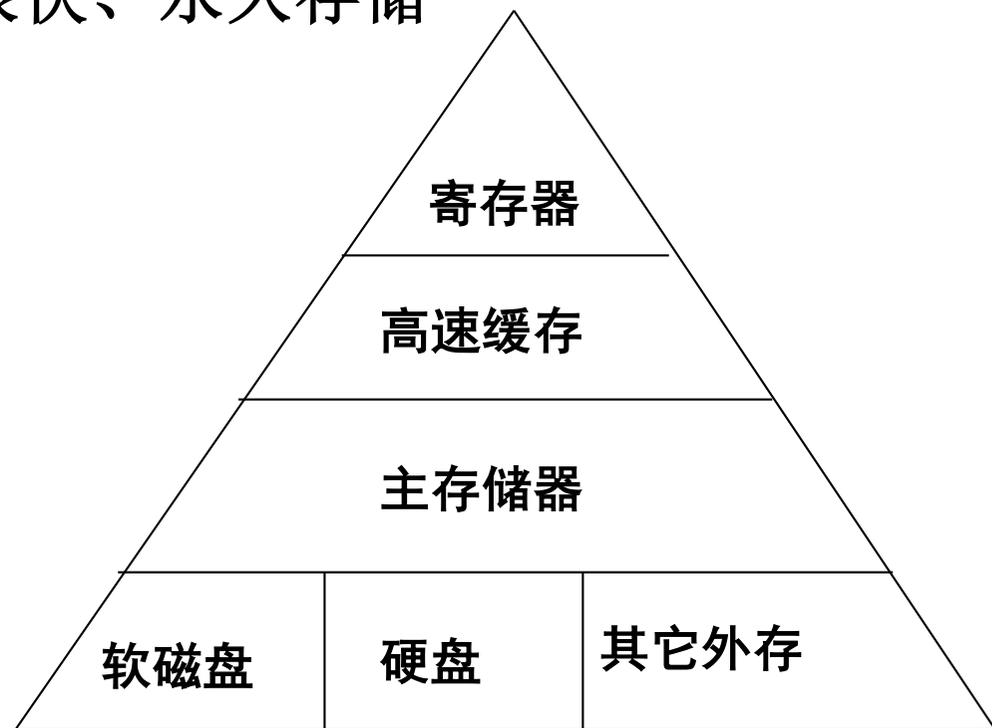
- 程序大小的增长速度比内存容量的增长快

□ 程序员的梦想

- 内存容量无限大、速度无限快、永久存储

□ 操作系统存储管理

- 充分利用内存，为程序并发执行提供存储基础
- 尽可能方便用户使用，如自动装入用户程序
- 解决程序空间比实际内存空间大的问题



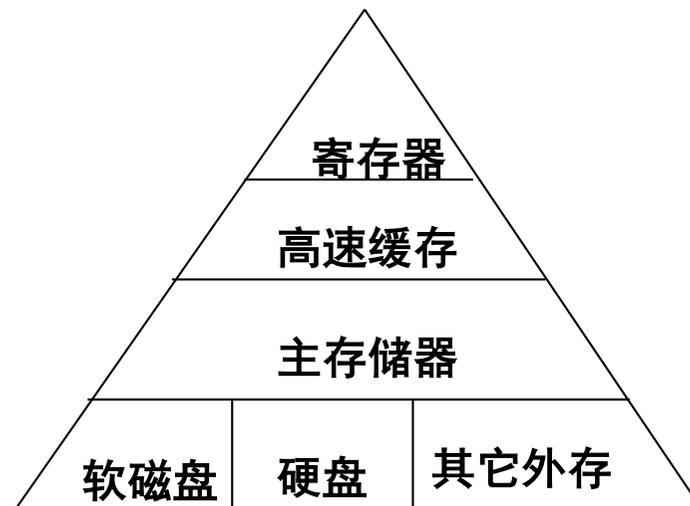
存储管理

□ 操作系统存储管理

- 记录内存空闲与否
- 为进程分配和释放存储空间
- 管理主存与外存间的数据交换

□ 存储管理技术分类

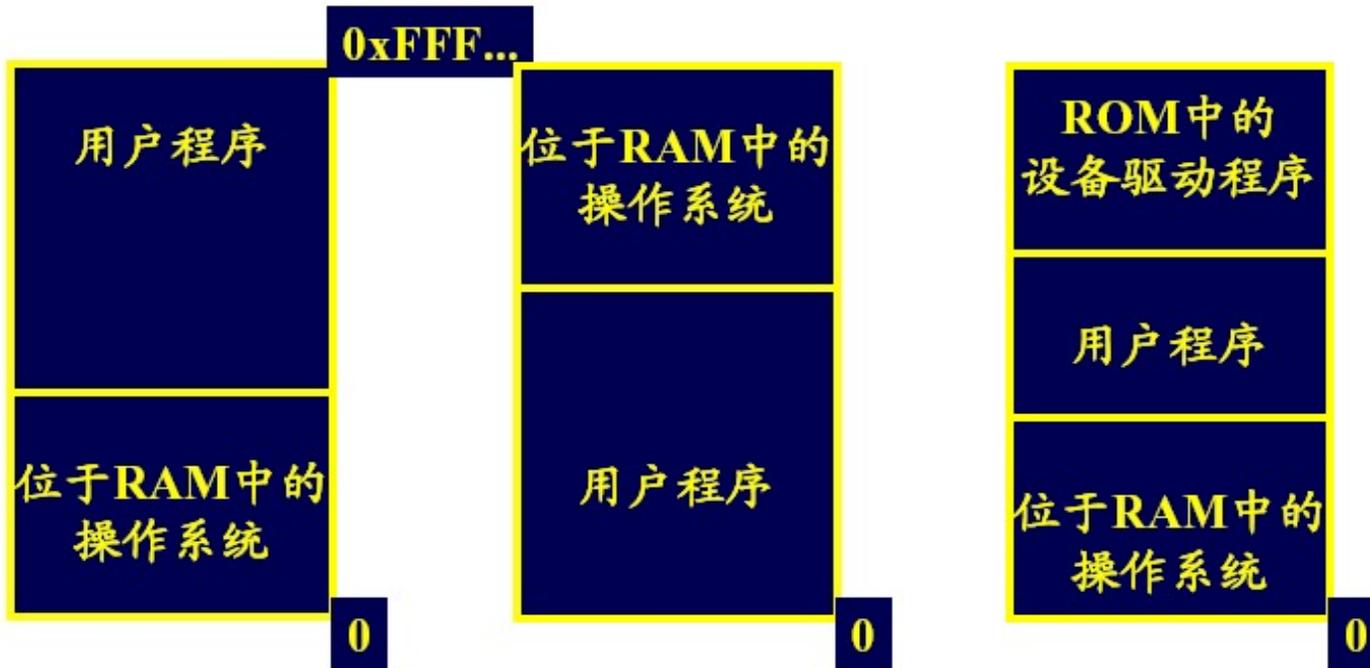
- 不在主存与磁盘间移动进程
 - 主存足以存储进程
 - 连续区 **vs.** 分区
- 在主存与磁盘间移动进程
 - 主存不足以存储进程
 - 交换 **vs.** 虚拟存储



0000	11101101	11101101	1000
0001	11001101	11101101	1001
0010	11101101	11001101	1010
0011	11101001	10001101	1011
0100	11101101	11101101	1100
0101	10001101	11101101	1101
0110	11101101	10001101	1110
0111	11101101	11101101	1111

单一用户（连续区）存储管理方案

- 单用户系统在一段时间内，只有一个进程在内存
 - 内存分配管理十分简单，内存利用率低



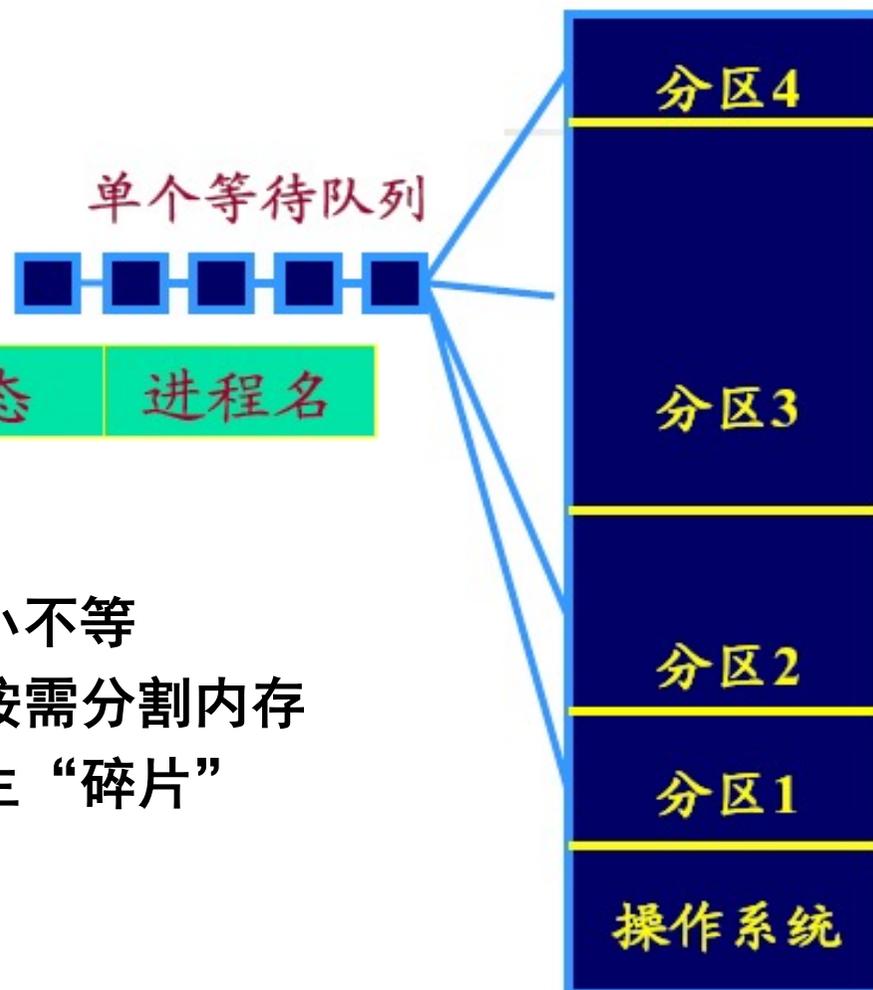
- 不在主存与磁盘间移动进程
 - 连续区 vs. 分区

- 在主存与磁盘间移动进程
 - 交换 vs. 虚拟存储

分区存储管理方案

支持多个程序并发

- 系统把内存用户区划分为若干分区
- 一个进程占据一个分区



分区号	起始地址	长度	状态	进程名
-----	------	----	----	-----

固定分区

- 分区大小相等
- 设置内存分配表
- 内存利用率不高

可变分区

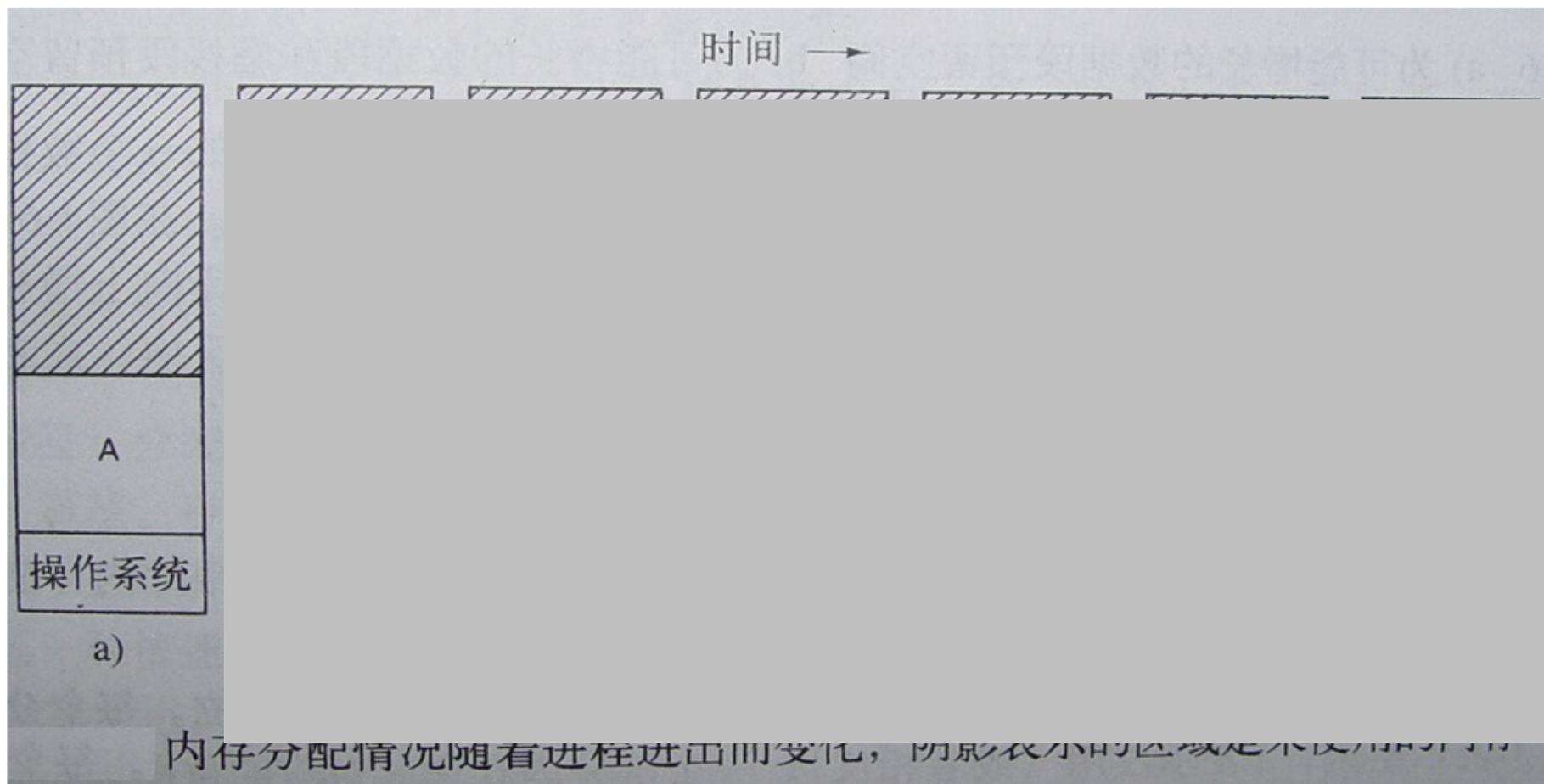
- 分区大小不等
- 分配时按需分割内存
- 容易产生“碎片”

- 不在主存与磁盘间移动进程
 - 连续区 vs. 分区

- 在主存与磁盘间移动进程
 - 交换 vs. 虚拟存储

交换存储管理方案

- ❑ 把暂时不运行的进程存到磁盘，在需要时调回主存运行



- 不在主存与磁盘间移动进程
 - 连续区 vs. 分区

- 在主存与磁盘间移动进程
 - 交换 vs. 虚拟存储

虚拟存储

□ 覆盖 (Overlay)

- 当一个程序太大时，程序员手工将程序分割成许多片断，操作系统完成这些片断的交换
- 这种手工分段的工作费时、枯燥、易错

□ 虚拟存储

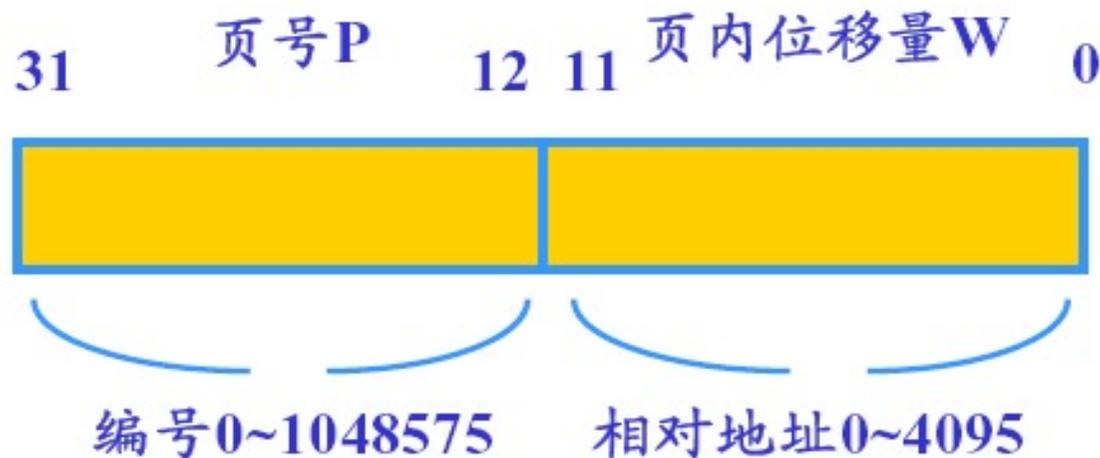
- 操作系统自动分割程序
- 对程序员而言，得到了比实际内存容量大得多的内存空间
 - 如现在的**Windows**，整个硬盘都可以做成虚拟主存
 - 在运行游戏或图像处理软件时，硬盘灯狂闪就是操作系统在交换数据
- 不在主存与磁盘间移动进程
 - 连续区 vs. 分区
- 在主存与磁盘间移动进程
 - 交换 vs. **虚拟存储**



页式存储管理方案

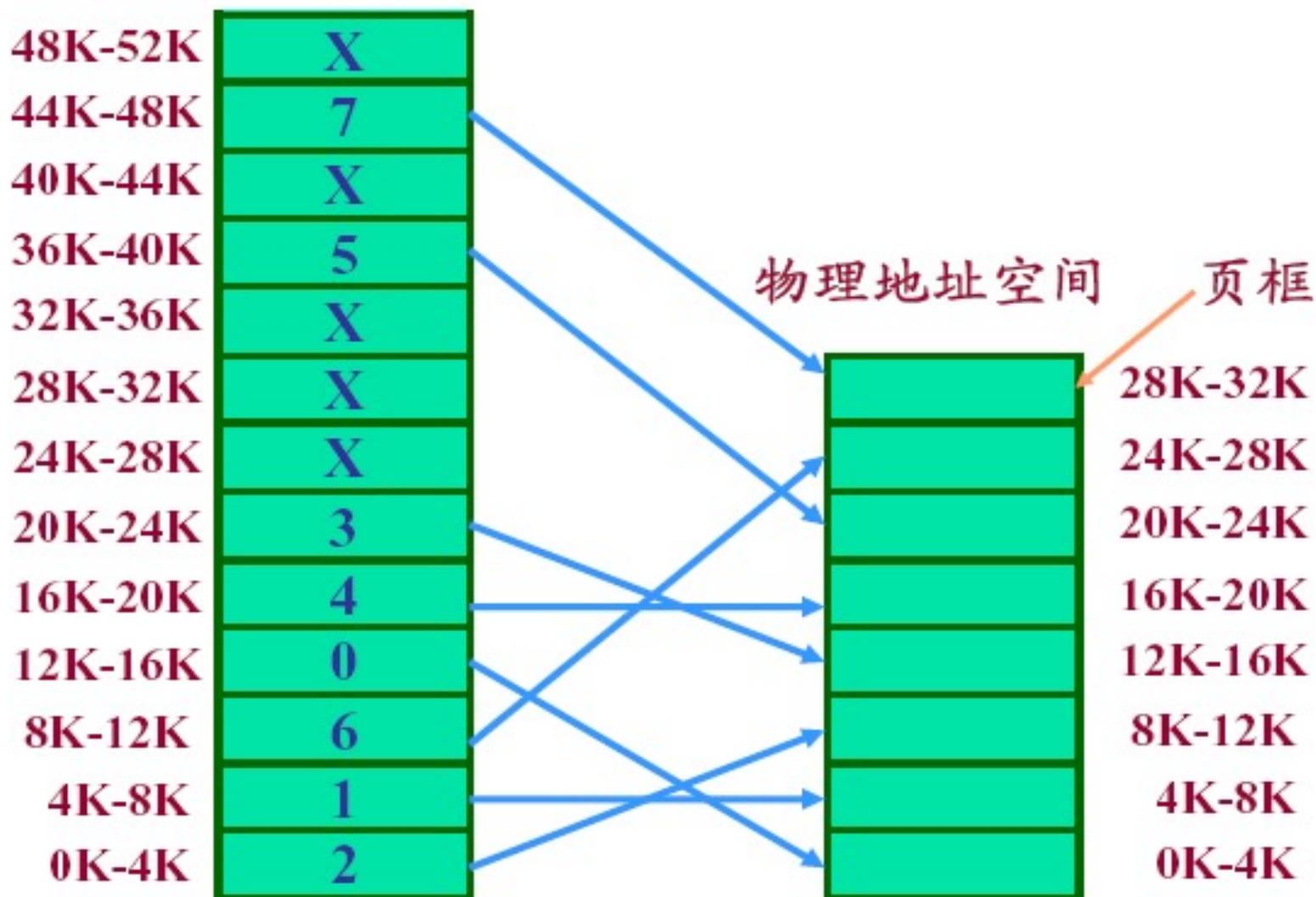
□ 基本思想（工作原理）

- 把用户程序按逻辑页划分成大小相等的部分
 - 称为页
- 从0开始编制页号，页内地址是相对于0编址
 - 逻辑地址



页式存储管理方案

逻辑（虚）地址空间



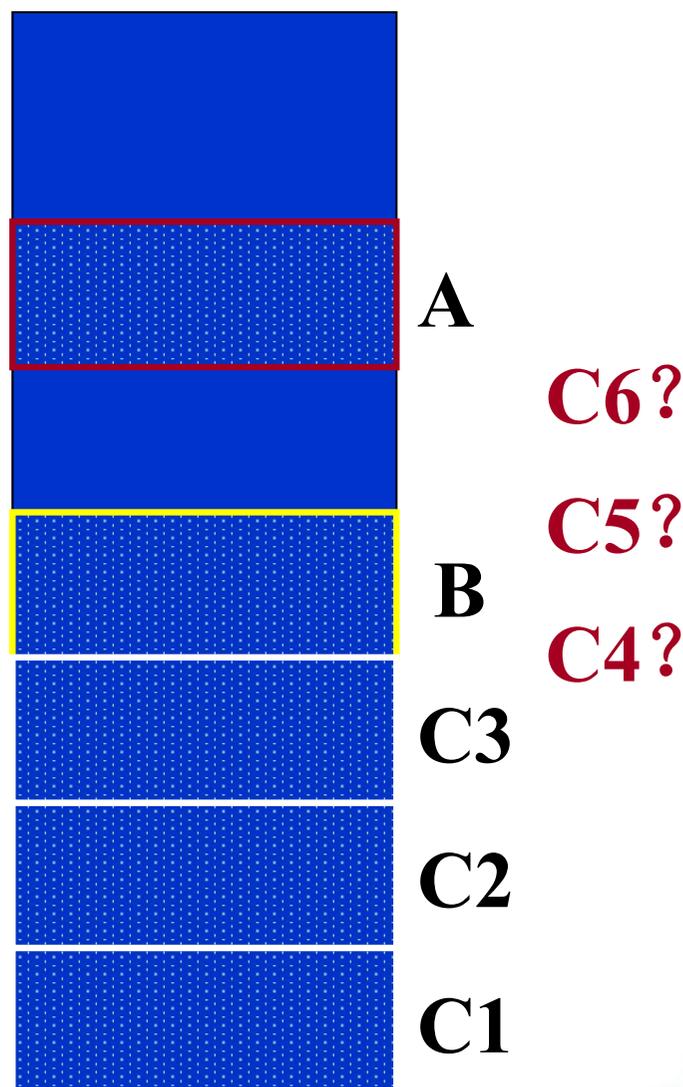
段式存储管理方案

□ 页式存储管理方案的不足

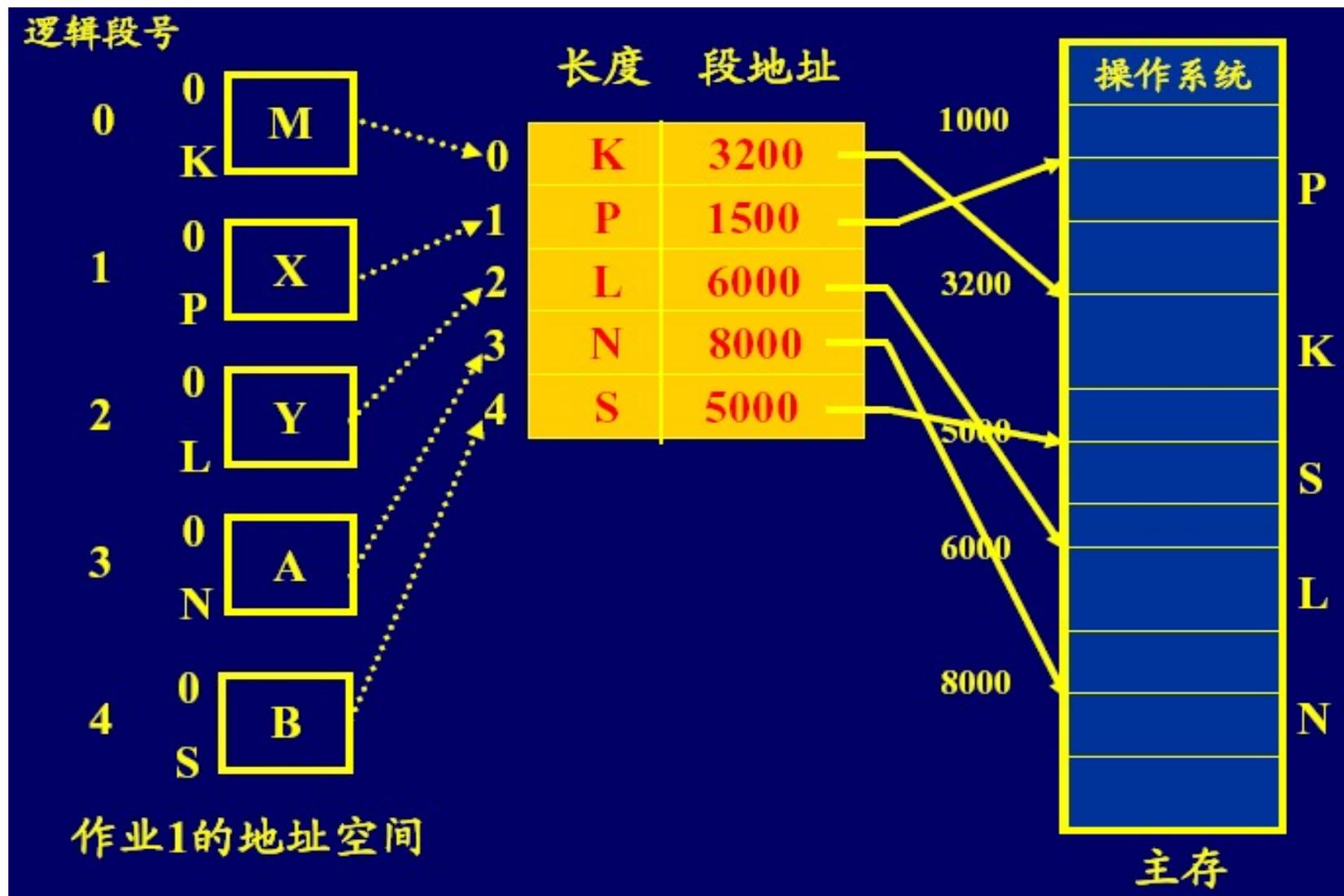
- 页的大小固定，不适合大小动态变化的数据块

□ 段式存储管理方案

- 把用户程序按逻辑关系划分成不定长的部分
 - 称为段
- 段一般而言足够大，且程序员只将一种类型的内容放在一个段中，因此，很少出现一个段容量不够的情况



段式存储管理方案



段页式存储管理方案

- 分段较大时，并不一定要全部放在主存
 - 也可能整个主存都放不下
- 自然地，对段进行分页
 - 对用户程序分段
 - 对内存分页

段号	段内地址	
	页号	页内地址

Windows 存储管理



Windows 任务管理器

文件(F) 选项(O) 查看(V) 关机(U) 帮助(H)

应用程序 进程 性能 联网 用户

CPU 使用
3 %

CPU 使用记录

PF 使用率
552 MB

页面文件使用记录

总数		物理内存 (K)	
句柄数	15746	总数	523496
线程数	561	可用数	57068
进程数	58	系统缓存	153412

认可用量 (K)		核心内存 (K)	
总数	566000	总数	89392
限制	1276464	分页数	69824
峰值	569684	未分页	19568

进程数: 58 CPU 使用: 3% 提交更改: 552M / 1246M



性能选项

视觉效果 高级 数据执行保护

处理器计划
默认情况下, 计算机设置为用较多的处理器时间来运行您的程序。
调整以优化性能:
 程序 (P) 后台服务 (S)

内存使用
默认情况下, 计算机设置为用较多的内存来运行您的程序。
调整以优化性能:
 程序 (R) 系统缓存 (T)

虚拟内存
页面文件存在于硬盘上, Windows 将它作为内存来使用。
所有驱动器页面文件大小的总数: 766 MB
[更改 \(C\)](#)

确定 取消 应用 (A)



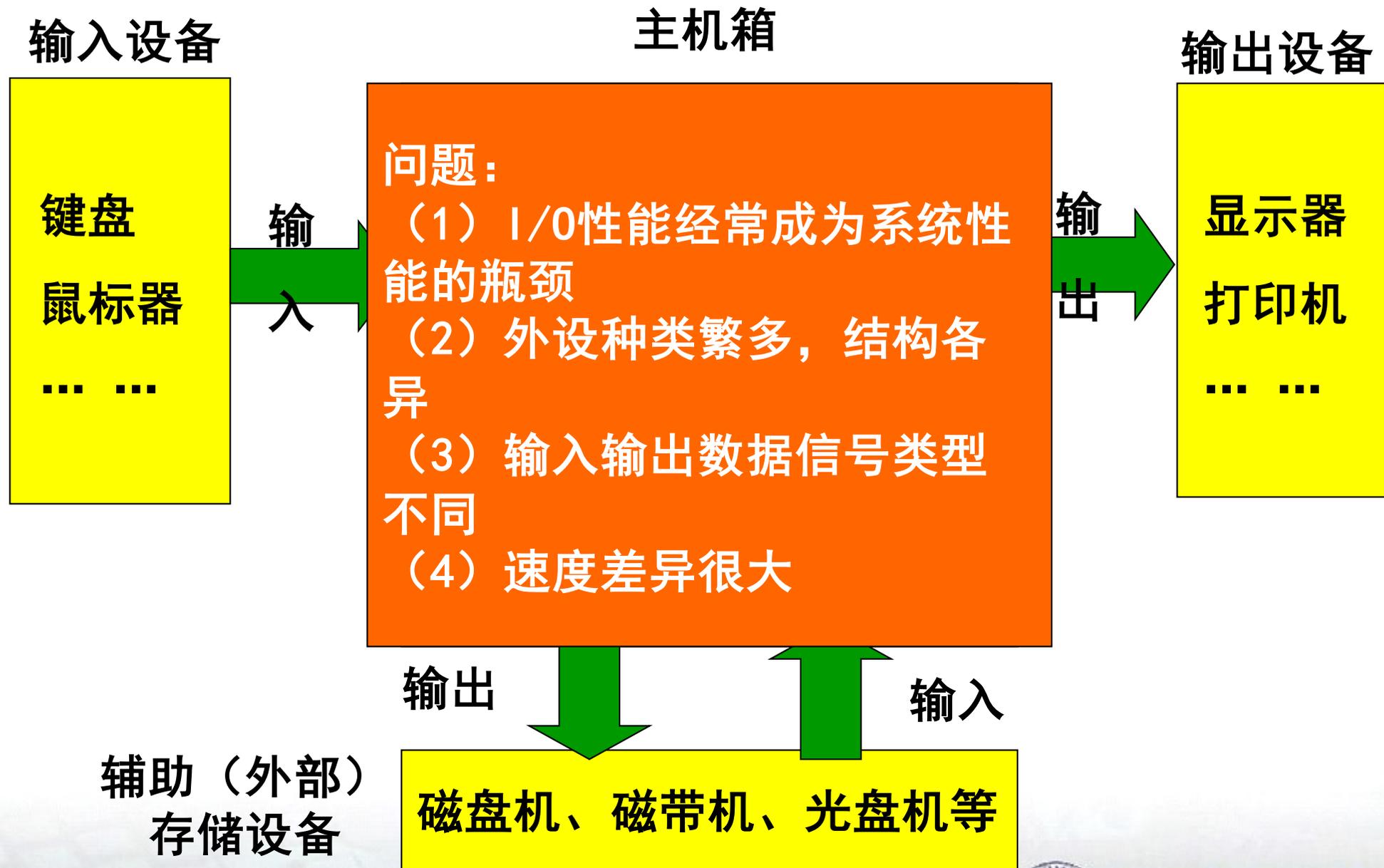
操作系统的功能



进程管理：
调度CPU和分配系统资源

存储管理：
内存空间的管理、分配与回收、共享和保护、内存扩充、地址转换

为什么要设备管理



为什么要设备管理

□ 完成用户的I/O请求

- 按照用户的请求，控制设备的各种操作，完成I/O设备与内存之间的数据交换
- 包括设备分配与回收；设备驱动程序；设备中断处理；缓冲区管理

□ 方便的编程接口、设备独立性

- 逻辑设备与物理设备、屏蔽硬件细节（设备的物理细节，错误处理，不同I/O的差异性）
- 用户能独立于具体物理设备而方便的使用设备

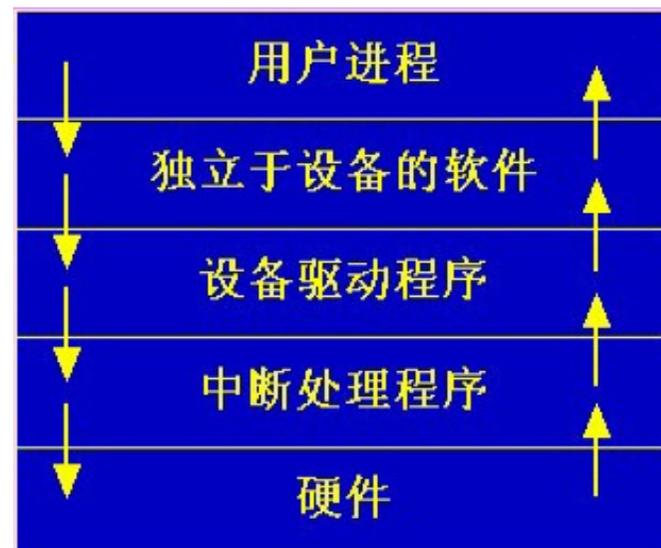
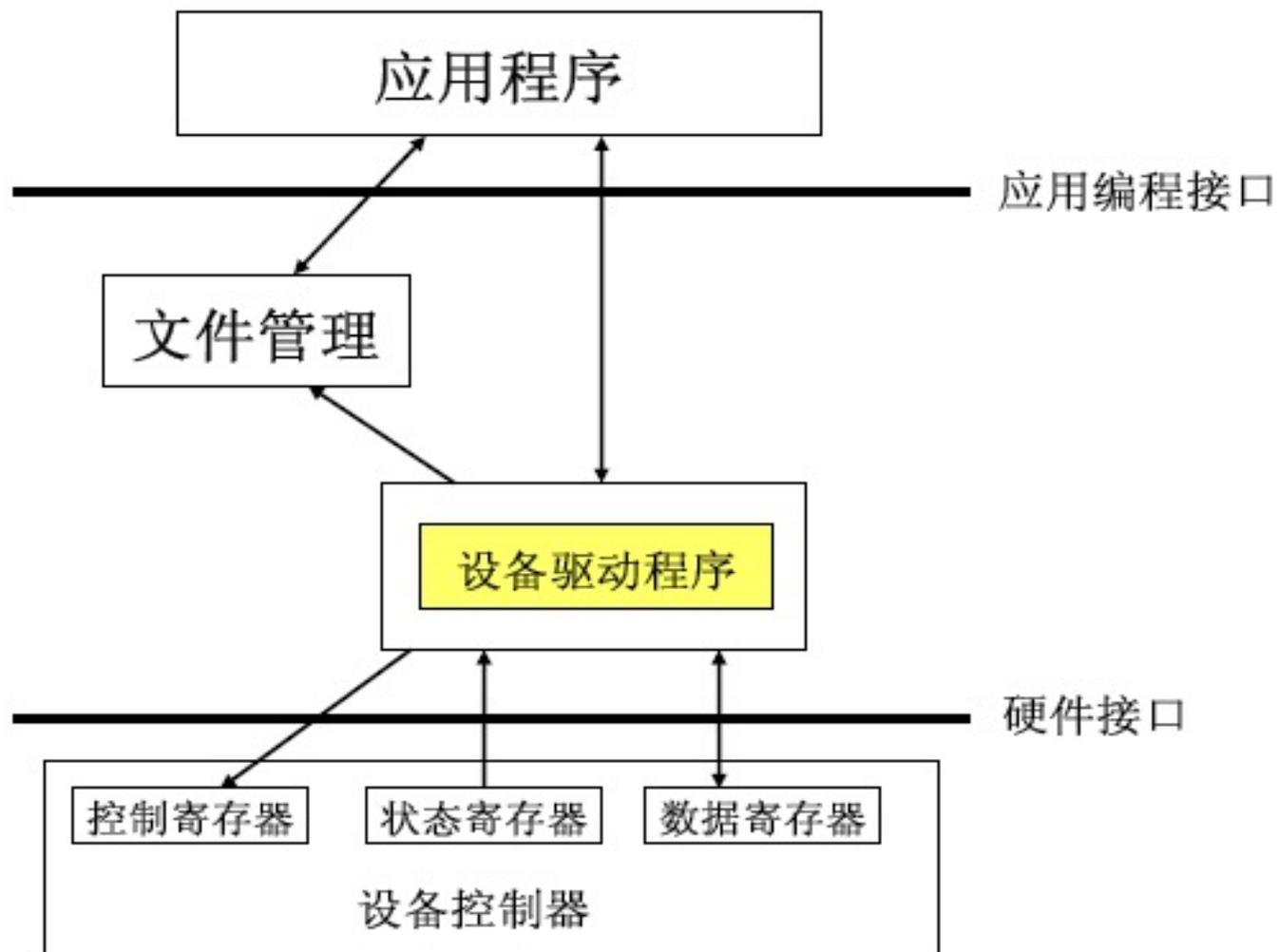


为什么要设备管理

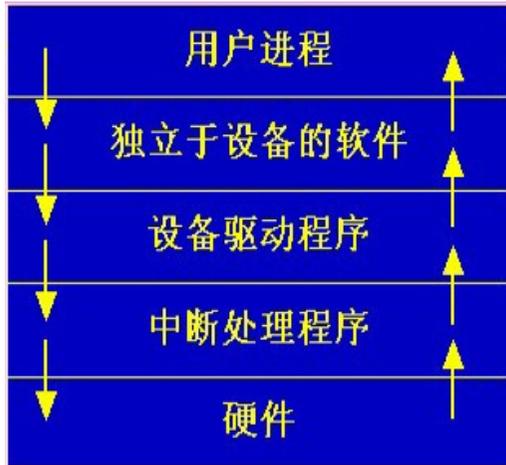
- ❑ 完成用户的I/O请求
- ❑ 方便的编程接口、设备独立性
- ❑ 协调多个进程间的竞争、提高资源利用率
 - 保证在多道程序环境下，当多个进程竞争使用设备时，按一定策略分配和管理各种设备，使系统能有条不紊的工作
 - 充分利用各种技术（通道，中断，缓冲等）提高CPU与设备、设备与设备之间的并行工作能力
- ❑ 保护设备传送或管理的数据（安全、无损、保密）



设备管理



设备管理



- (1) 用户进程层执行输入输出系统调用，对I/O数据进行格式化，为假脱机输入/输出作准备
- (2) 独立于设备的软件实现设备的命名、设备的保护、成块处理、缓冲技术和设备分配
- (3) 设备驱动程序设置设备寄存器、检查设备的执行状态
- (4) 中断处理程序负责I/O完成时，唤醒设备驱动程序进程，进行中断处理
- (5) 硬件层实现物理I/O的操作

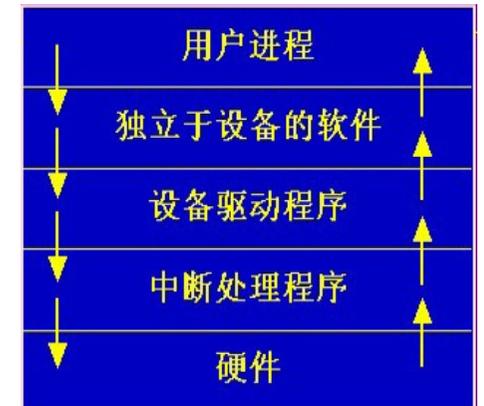
设备管理

□ 中断处理程序

- 通过中断请求(**interrupt requests, IRQ**)来维护**CPU**和硬件设备之间的通信
- 每个进程在启动一个**I / O**操作后阻塞
- 直到**I / O**操作完成并产生一个中断
- 由操作系统接管**CPU**后唤醒该进程为止

□ DMA (Direct Memory Access)

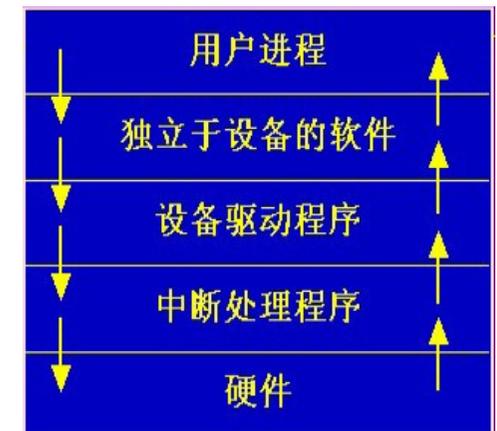
- 数据在内存与**I/O**设备间直接成块传送
- **CPU**在开始时向设备发“传送一块”命令，结束时进行相应处理
- 实际操作由**DMA**硬件直接完成



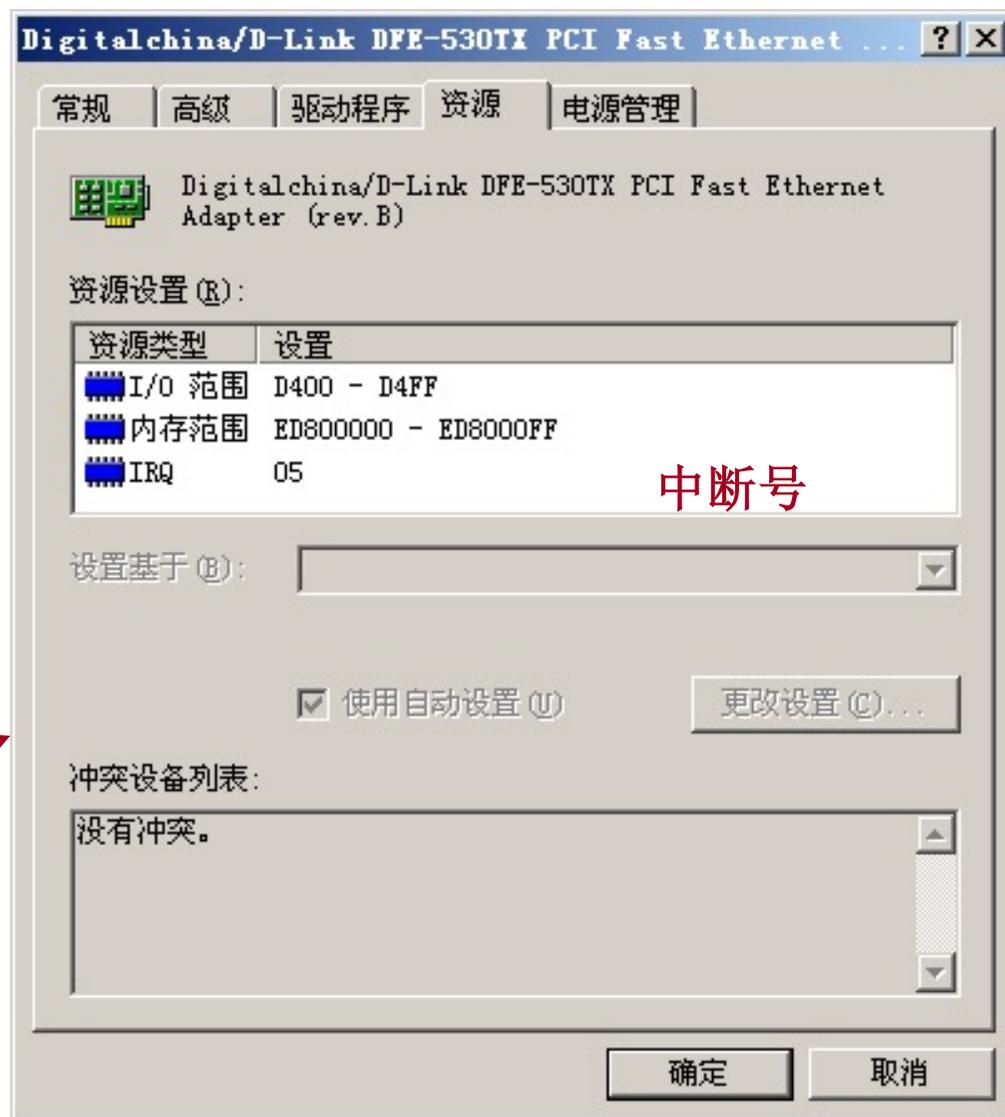
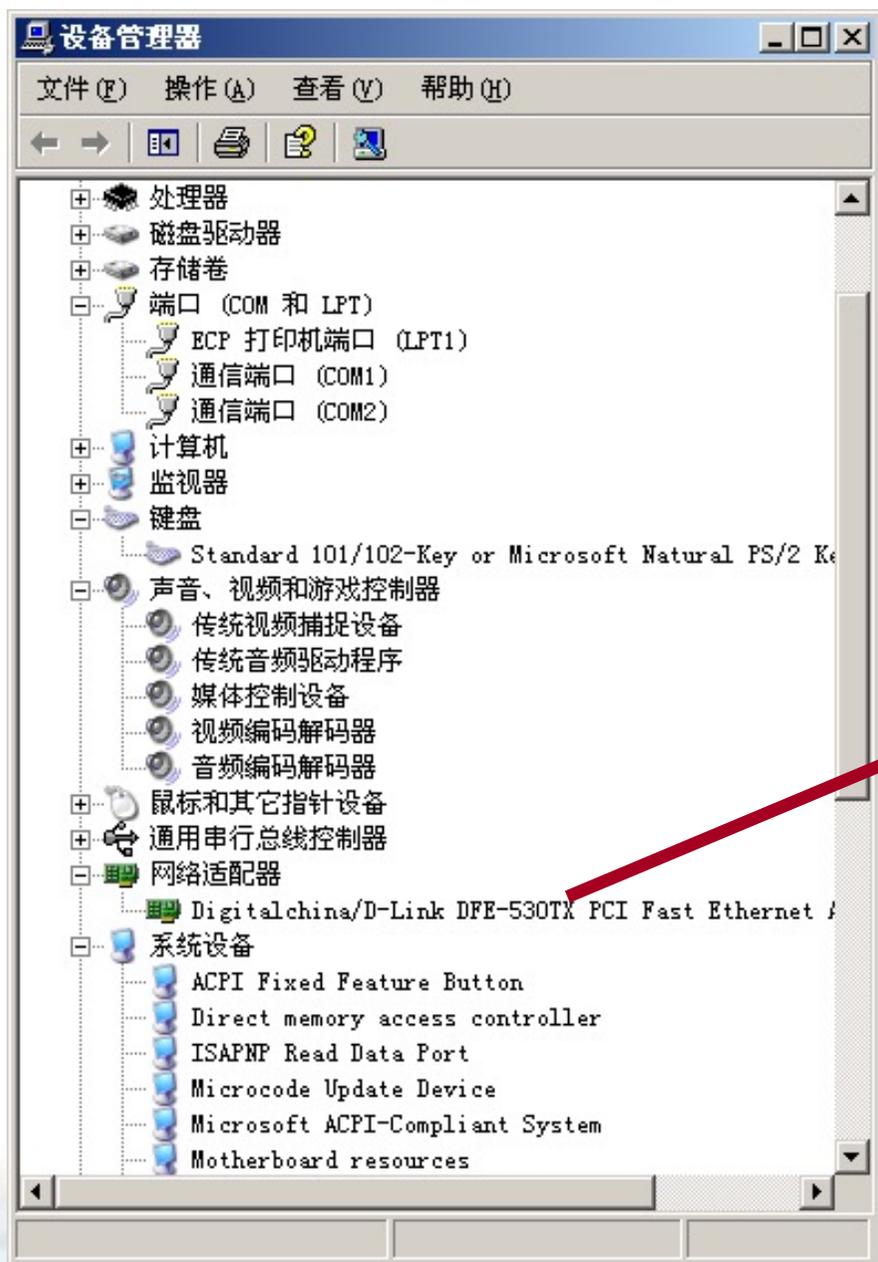
设备管理

□ 设备驱动程序

- 为了控制**I/O**传输，系统为每类设备编制设备驱动程序
- 主要负责接收和分析从设备分配转来的信息，并根据设备分配的结果，结合具体物理设备特性完成以下具体工作
 - (1) 预置设备的初始状态
 - (2) 根据请求传输的数据量，组织**I/O**缓冲队列，利用**I/O**缓冲对数据进行加工，包括数据格式处理和编码转换
 - (3) 构造**I/O**程序
 - (4) 启动设备进行**I/O**操作



Windows设备管理



即插即用技术

□ Plug and Play, PnP

- 在计算机内插入一个设备并使计算机确认此设备的存在，而用户不必通知计算机
- “对号入座”
 - 事先为各类硬件发一张身份证，然后系统根据不同硬件的身份证号分配系统中的**IRQ**、**I/O**等资源，并自动搜索、加载硬件的驱动程序
- 即插即用的实现需要硬件厂商的生产遵循特定标准，同时操作系统提供支持

□ 真的“想用就用”？

- 不行，因为硬件资源有限，存在冲突
- **Universal PnP**

操作系统的功能



进程管理：
调度CPU和分配系统资源

存储管理：
内存空间的管理、分配与回收、共享和保护、内存扩充、地址转换

设备管理：
分配和回收外部设备以及控制外部设备按用户程序的要求进行操作

为什么需要文件管理

□ 所有的计算机应用程序都要：

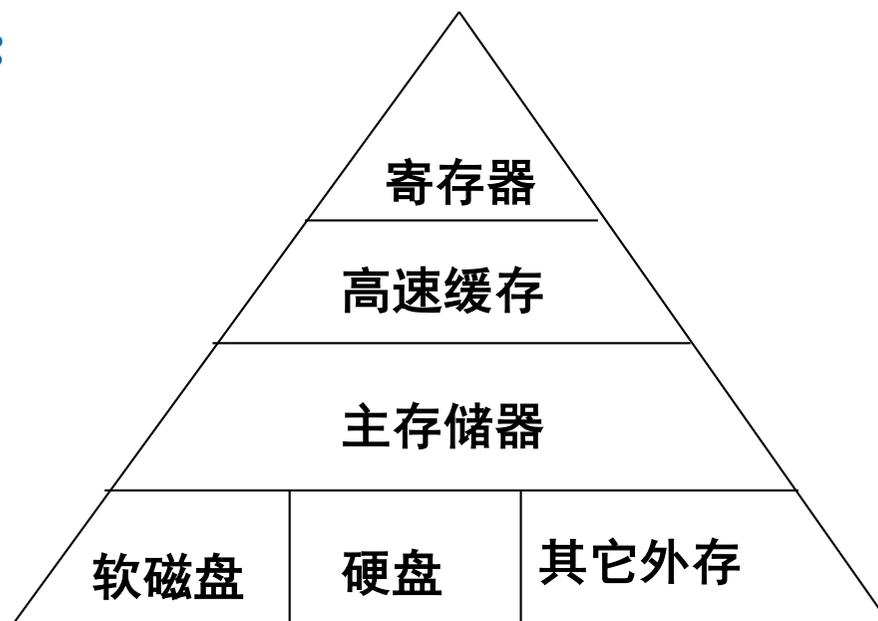
- 存储信息
- 检索信息

□ 三个基本要求：

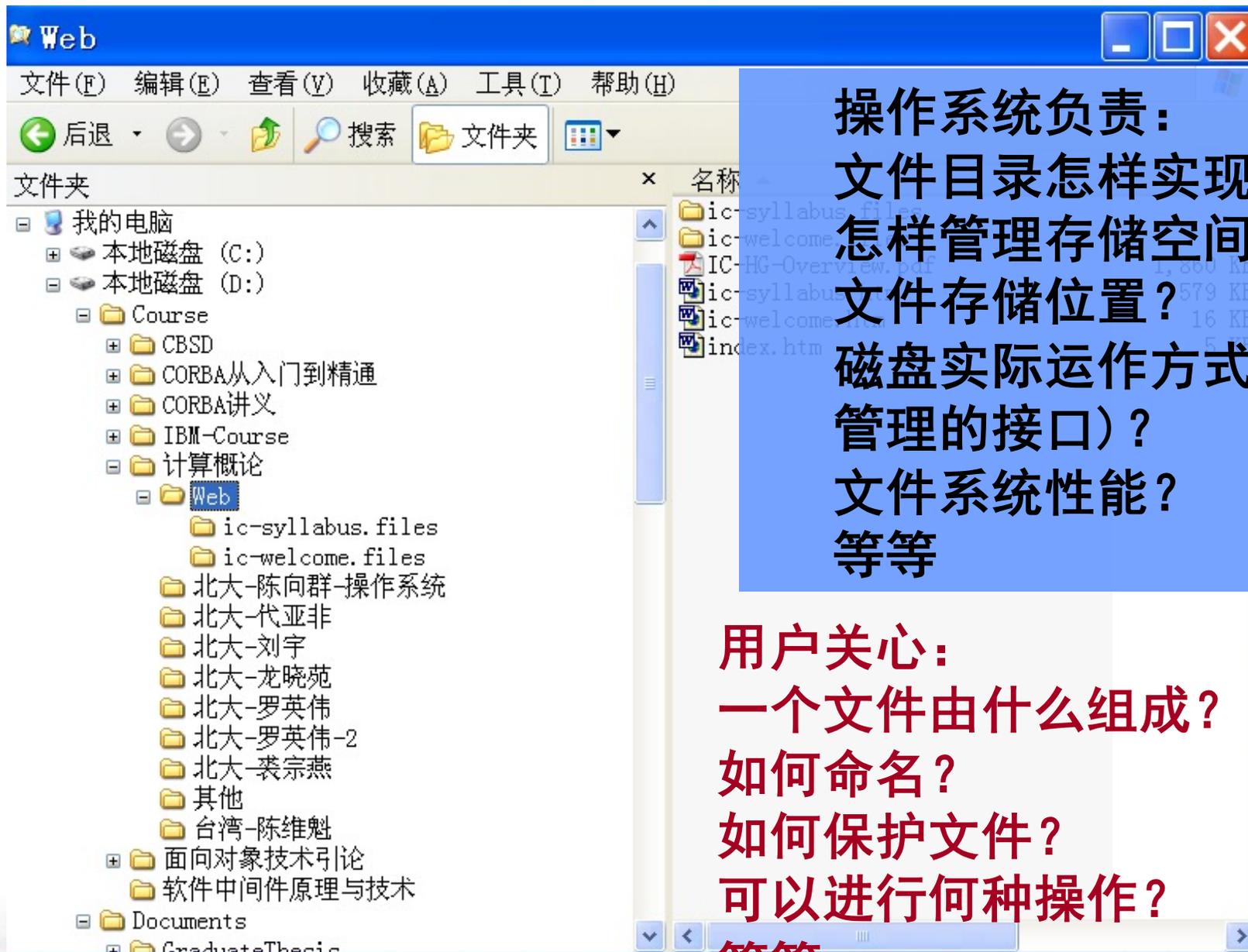
- 能够存储大量的信息
- 长期保存信息
- 可以共享信息

□ 解决方法

- 把信息以一种单元，即文件的形式存储在磁盘或其他介质上



文件与文件系统



操作系统负责：
文件目录怎样实现？
怎样管理存储空间？
文件存储位置？
磁盘实际运作方式（与设备管理的接口）？
文件系统性能？
等等

用户关心：
一个文件由什么组成？
如何命名？
如何保护文件？
可以进行何种操作？
等等



操作系统文件管理

□ 文件

- 一组带标识的在逻辑上有完整意义的信息项的序列

□ 文件系统

- 操作系统中统一管理信息资源的一种软件
- 管理文件的存储、检索、更新，提供安全可靠的共享和保护手段，并且方便用户使用需要处理

存储介质	磁带	磁盘		
物理结构	连续结构	连续	链接	索引
存取方式	顺序存取	顺序	顺序	顺序
		随机		随机

更多内容将在程序设计中介绍

Windows文件系统

	FAT 16	FAT 32	NTFS
支持的磁盘大小	2GB	32GB	2TB(2048GB)
文件最小单元(簇)的大小	32KB	4KB(2GB~8GB)	4KB

□ NTFS其他特性

- 支持文件加密和分别管理功能
- 支持对分区、文件夹和文件的压缩
- 可以为共享资源、文件夹以及文件设置访问许可权限
- 可以为用户所能使用的磁盘空间进行配额限制
- 可恢复（拷贝、粘贴、移动失败后可以自动恢复）

操作系统的功能



进程管理：
调度CPU和分配系统资源

存储管理：
内存空间的管理、分配与回收、共享和保护、内存扩充、地址转换

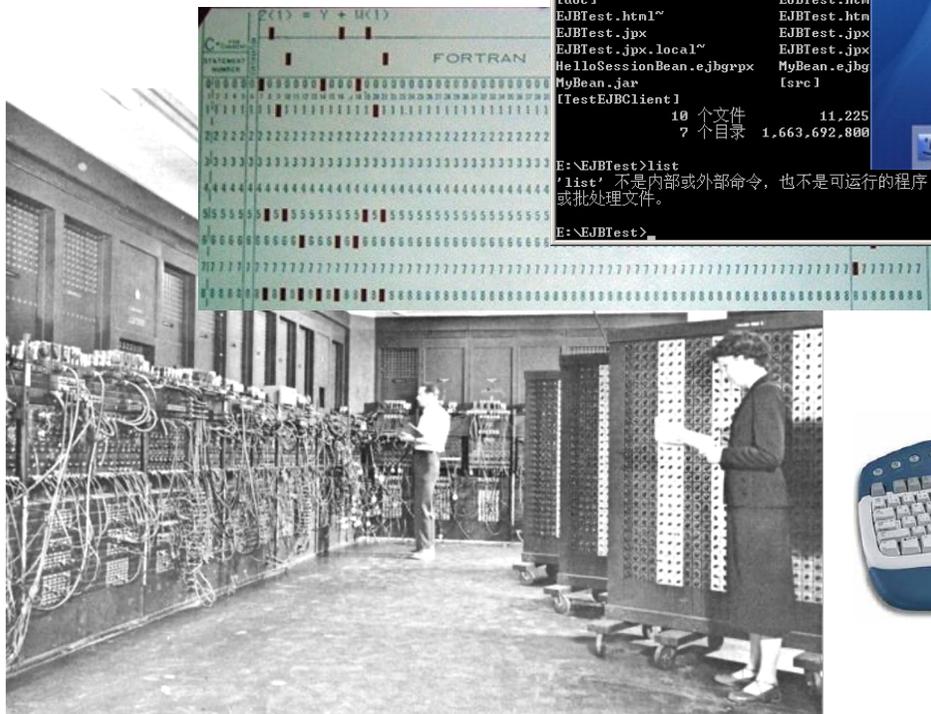
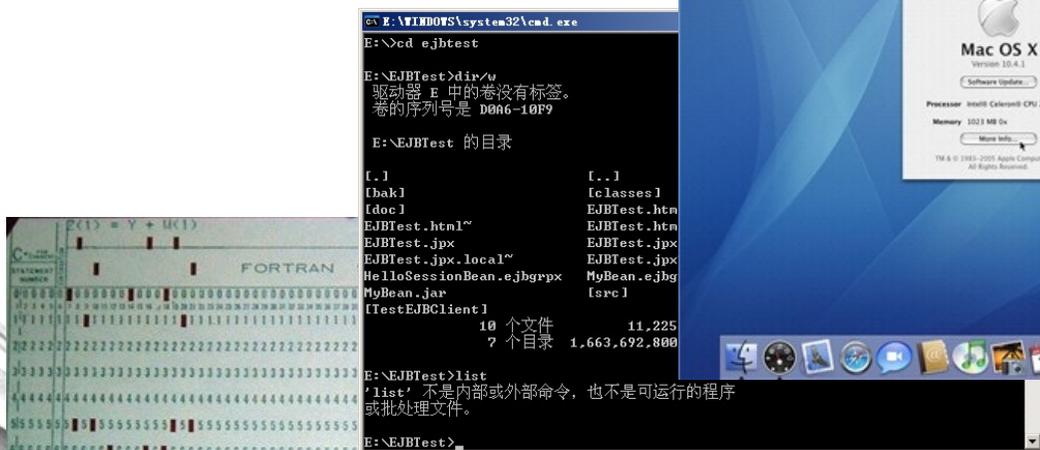
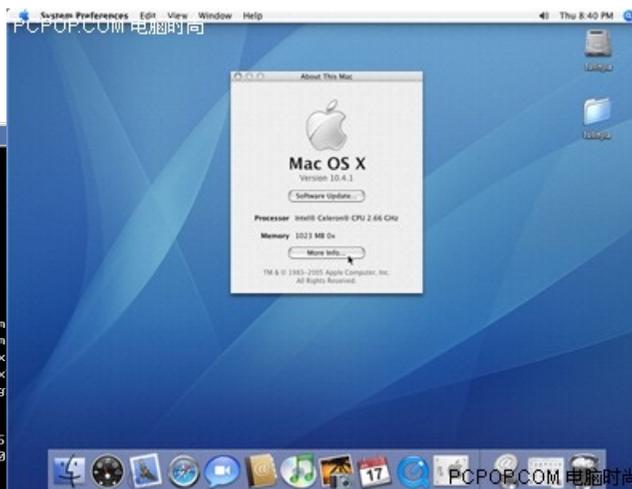
设备管理：
分配和回收外部设备以及控制外部设备按用户程序的要求进行操作

文件管理：
向用户提供创建、撤销、读写、打开、关闭文件等功能

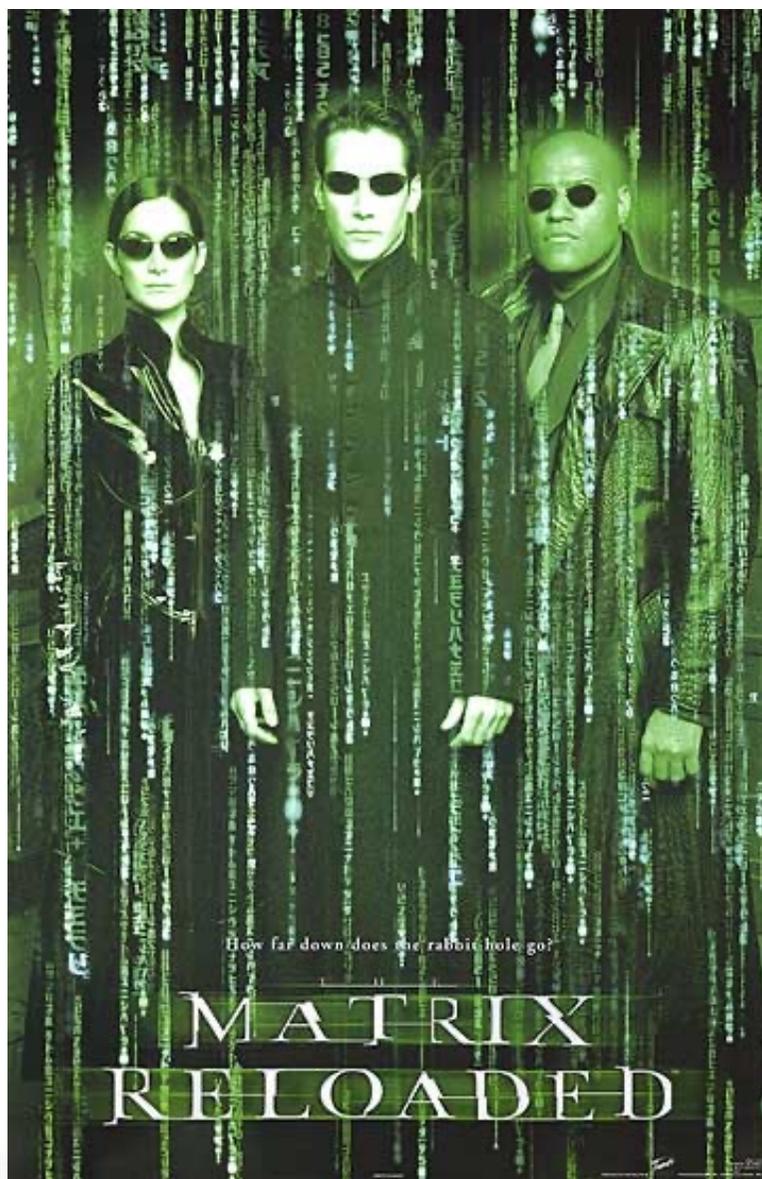
用户界面的演变

让用户方便、有效地使用计算机

→ 软硬件协同发展



未来的用户界面/人机交互



沉浸计算/虚拟现实

vs

泛在计算/普适计算

操作系统小结



进程管理：
调度CPU和分配系统资源

存储管理：
内存空间的管理、分配与回收、共享和保护、内存扩充、地址转换

设备管理：
分配和回收外部设备以及控制外部设备按用户程序的要求进行操作

用户管理：
提供一个友好的用户访问操作系统的接口

文件管理：
向用户提供创建、撤销、读写、打开、关闭文件等功能

