计算概论 之 认识计算机

系统软件

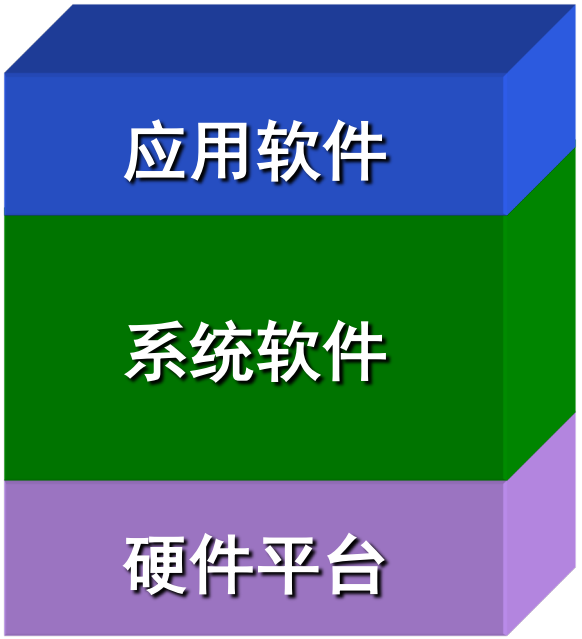
刘譞哲
北京大学计算机学院

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

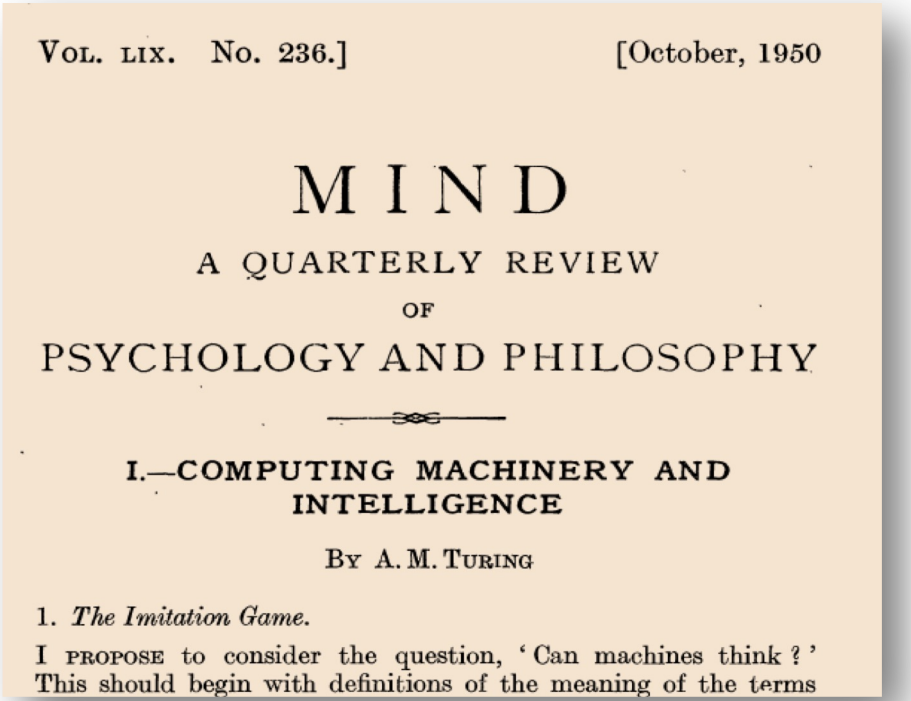
- 计算机软件可以分为 系统软件 和 应用软件
  - 系统软件：与应用领域无关，为其他软件提供基础设施支撑
    - 操作系统：Windows，MacOS，iOS
    - 程序语言环境：C语言，Basic语言，汇编语言
    - 数据库：Access，IBM DB2，Oracle
  - 应用类
    - 各类领域、行业的应用软件
      - 办公领域：MS Office、WPS
      - 设计软件：PS，…..
      - 社交软件：抖音，微信….
      - 游戏软件
      - ……..



操作系统在计算机发展历史上的重要地位

- 在过去70余年间，计算机发生了巨大的变化
  - 但是，初心未变！
  - 图灵的初心(1950)：让计算机达到像人类一样的智能：图灵测试
  - 当前的人工智能只是达到的一个阶段性目标

- 计算机：从专用到通用
  - 操作系统的出现才使得计算走向通用，走向大众，带来计算机应用的繁荣！



VOL. LIX. No. 236.] [October, 1950

MIND

A QUARTERLY REVIEW

OF

PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND INTELLIGENCE

BY A. M. TURING

1. *The Imitation Game.*

I PROPOSE to consider the question, 'Can machines think?' This should begin with definitions of the meaning of the terms

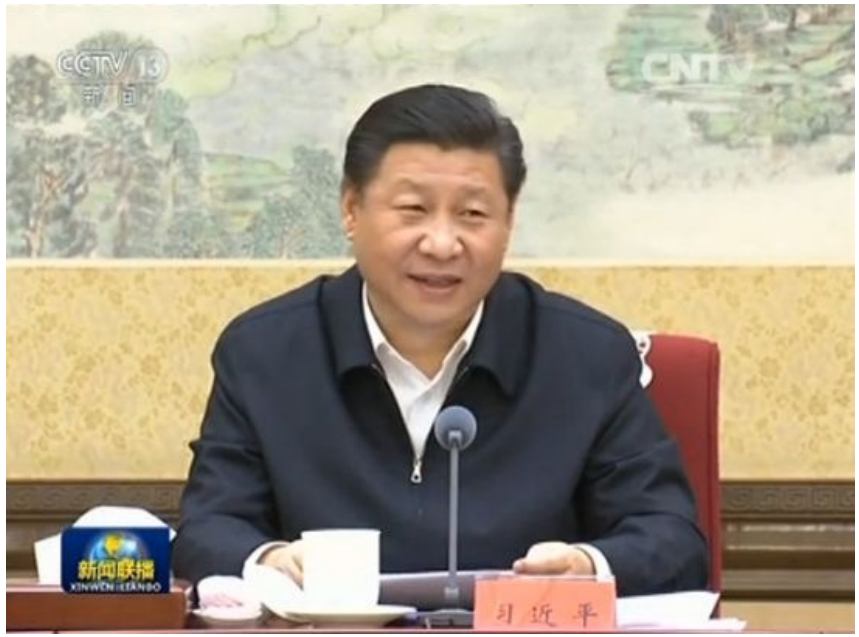
Turing, Alan (1950), "Computing Machinery and Intelligence", *Mind*, LIX (236): 433–460

操作系统在计算机技术体系的重要地位

❑ **操作系统和CPU 是计算机系统的核心**
也是产业生态的核心
更是信息时代安全的基石

- Wintel（Windows + Intel）形成了桌面计算时代事实上的垄断，占领了90%的PC市场
- 1999年数据：比尔.盖茨拥有540亿美元财富，每周以4亿美元速率增长。微软公司产值超过美国三大汽车公司产值总和

操作系统在国家发展中的重要地位



国家对操作系统给予了高度关注和投入

习总书记：

“核心技术靠化缘是要不来的，只有自力更生”、“实施网络信息领域核心技术设备攻坚战略，推动高性能计算、移动通信、量子通信、核心芯片、操作系统等研发和应用取得重大突破”

——2016年10月9日在十八届中央政治局第三十六次集体学习时的讲话

亟需打破操作系统“缺芯少魂”的局面

- 基础软件作为应用软件的运行和开发平台，对国家安全更是起着决定性的作用！
- 但是，“空心化”和“低端化”仍然是我国信息技术发展的核心问题
- CPU、操作系统、数据库管理系统等基础软硬件长期缺失

Microsoft 微软长期垄断PC操作系统

2008年10月20日起，微软正式对使用盗版Windows XP的用户强制实施正版验证和“黑屏”提示。如果用户不能通过验证，Office软件上将被永久添加视觉标记，桌面背景每隔1小时就要被“黑屏”一次！

亟需打破操作系统“缺芯少魂”的局面

移动操作系统，仍然是苹果和谷歌“二分天下”

iOS

苹果封闭系统形成了完整的生态链，为苹果获取了的巨大利润

ANDROID

- **谷歌Android通过开源建立了生态系统，装机量占全球移动端操作系统的60%以上**
- **我国手机产量全球第一，华为、OPPO和小米手机销量仅次于三星和苹果，但完全依赖谷歌Android生态系统！**

2019年5月20日，谷歌母公司Alphabet宣布停止与华为公司相关的业务和服务，涉及硬件、软件和技术服务方面的合作。华为将失去对谷歌Android操作系统的更新访问权限，其海外用户将受到影响！

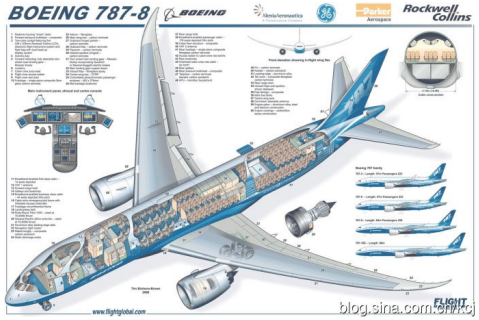
华为突遭谷歌釜底抽薪：官方安卓不再支持华为手机

量子位

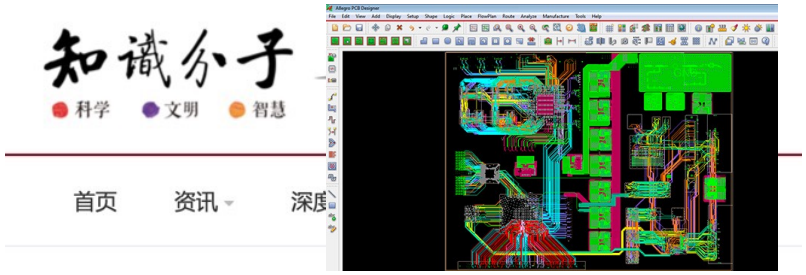
2019-05-20 08:48 收藏11 评论32 社交通讯

亟需打破操作系统 “缺芯少魂”的局面

缺乏自主的工业软件特别是工业操作系统，核心技术和数据面临被国外控制的巨大风险，已成为“卡脖子”之痛！



7000多种



cadence

Cadence在芯片设计自动化领域领先者和垄断者，在2018年率先响应美国政府号召对中芯事件实施禁运。没有它的授权，很难设计芯片！

国内一些制造企业的后台核心操作系统，以及关键的软件工具如CAD（计算机辅助设计）和CAE（计算机辅助仿真）都来自国外，比如西门子、SAP和GE，后续的技术改进只能依赖国外厂商进行二次开发！

操作系统的定义

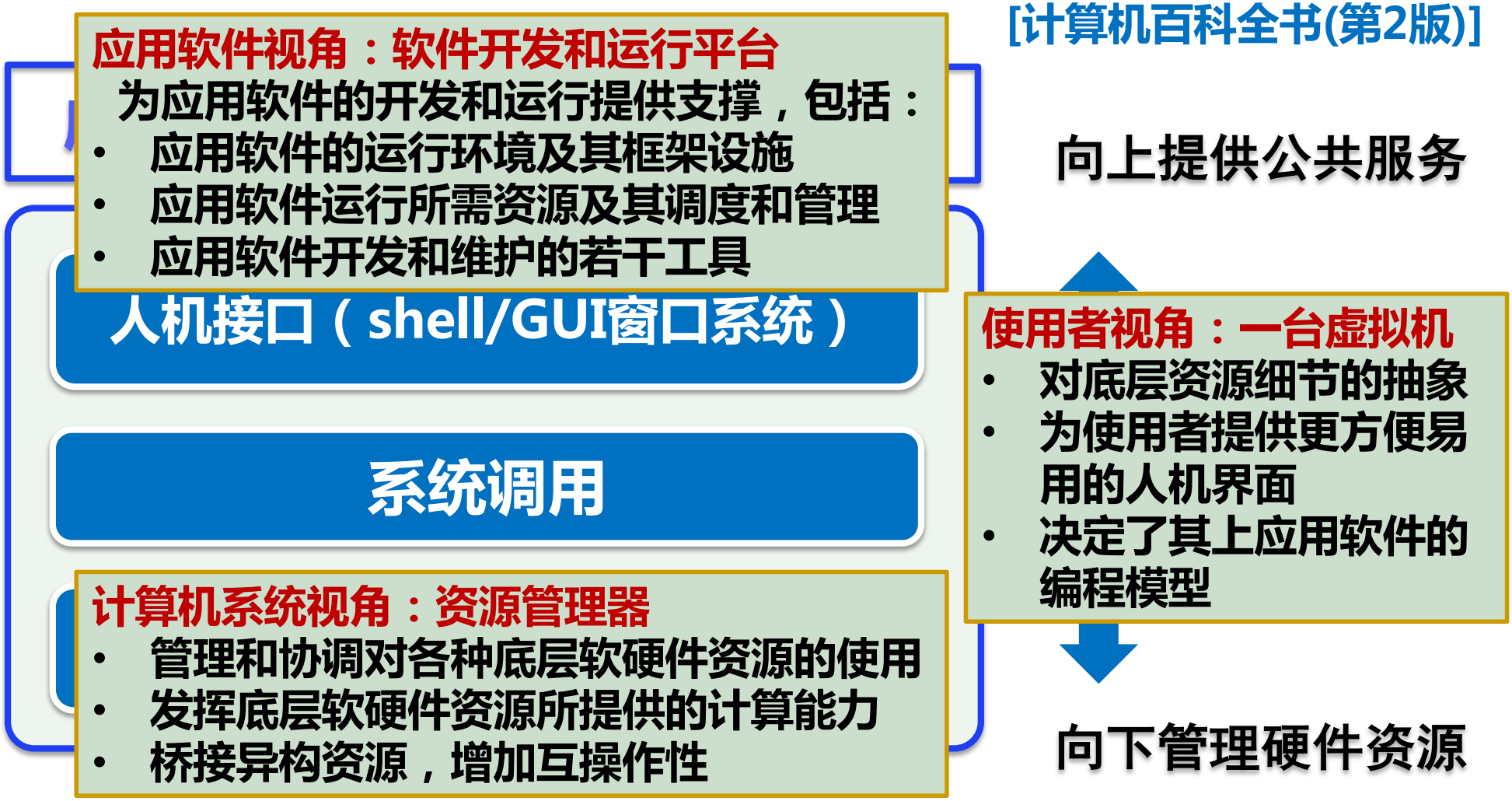
- 操作系统是管理硬件资源、控制程序运行、改善人机界面和为应用软件提供支持的一种系统软件。

[计算机百科全书(第2版)]

操作系统的定义

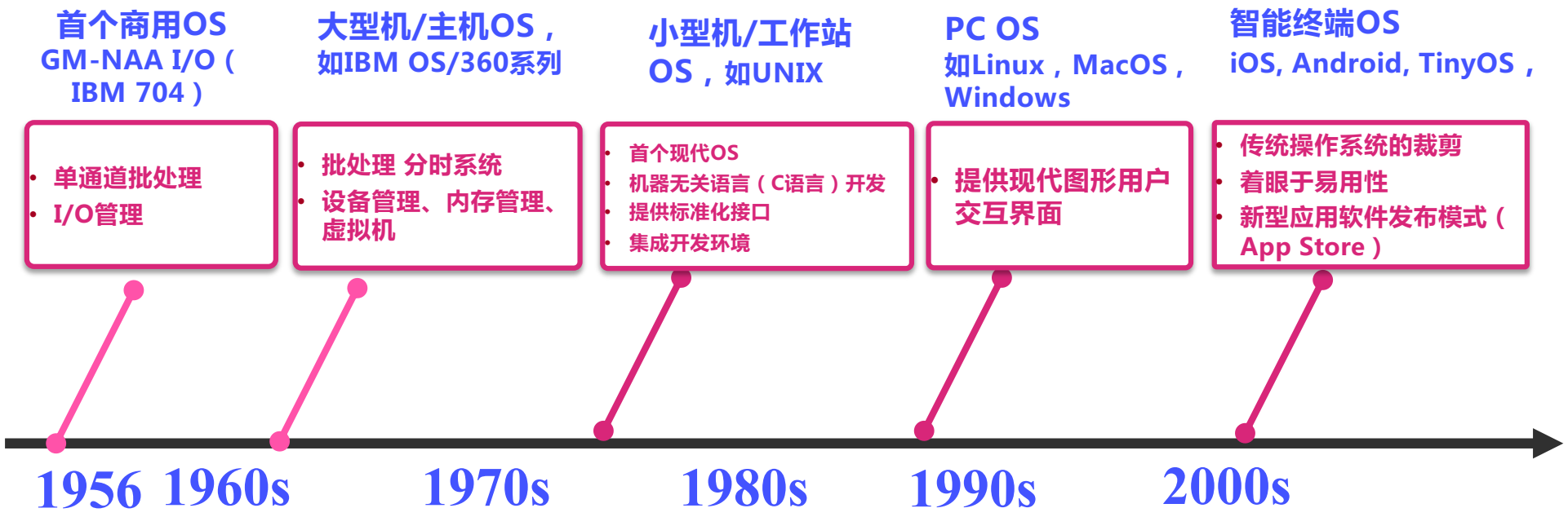
- 操作系统是管理硬件资源、控制程序运行、改善人机界面和为应用软件提供支持的一种系统软件。

[计算机百科全书(第2版)]



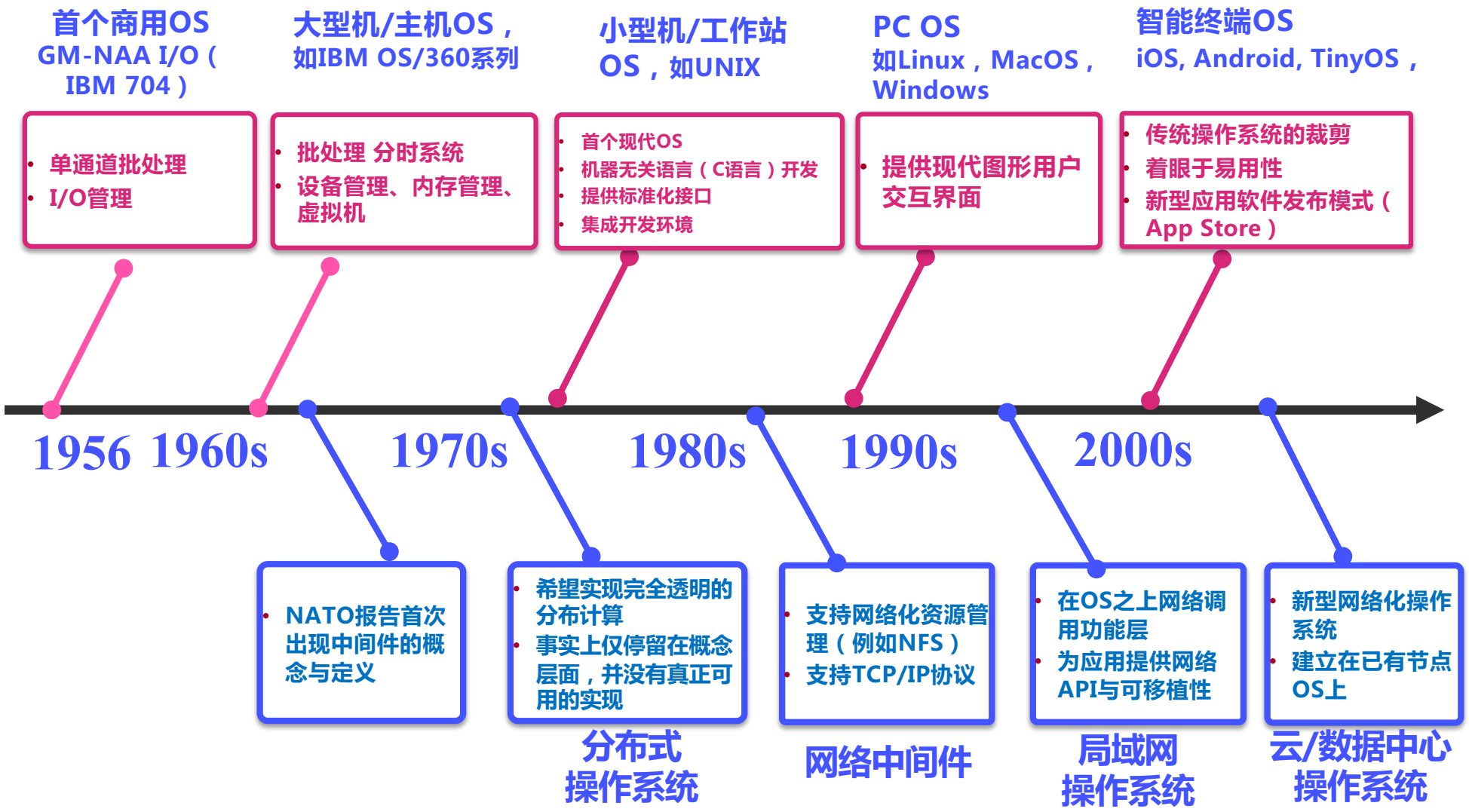
操作系统的发展历程回顾

主线：面向单机的节点OS，已基本定型



操作系统的发展历程回顾

主线：面向单机的节点OS，已基本定型



辅线：支持多机、网络、分布的OS，逐步成为创新焦点

操作系统发展的周期性规律

操作系统发展存在周期律

每20年左右出现一次跨越式发展机遇

计算环境	主机时代 （1960s-80s） IBM: OS/360 UNIX	PC时代 （1980s-2000s） Microsoft： Windows Linux	互联网时代 （2000s-20s） Google:Android Apple: iOS	人机物融合时代 （2020s--）
应用场景规模	国家 军事、科学计算 $10^1 \sim 10^3$	大企业 企业办公 $10^4 \sim 10^6$	个人 娱乐、电商 $10^7 \sim 10^9$	人、机、物 万物互联、智能应用 $10^{10} \sim 10^{12}$

贝尔定律：计算设备约每10年左右一次换代，设备、用户增加10倍，形成新的蓝海，催生新型应用，间次推动操作系统换代，形成新的生态

每当计算环境（赛道）发生变化，应用场景规模呈数量级增长

操作系统发展的周期性规律

操作系统发展存在周期律

每20年左右出现一次跨越式发展机遇

计算环境	主机时代 （1960s-80s） IBM: OS/360 UNIX	PC时代 （1980s-2000s） Microsoft： Windows Linux	互联网时代 （2000s-20s） Google:Android Apple: iOS	人机物融合时代 （2020s--）
基础支撑	分时、PV理论、虚拟化、多任务多用户....	ADT、面向对象、交互分时	TCP/IP、分布式、万维网.....	时空逻辑、容器、微服务.....


E. Dijkstra
1972年图灵奖


J. Backus
1977年图灵奖


K. Thompson & D. Richie
1983年图灵奖


F. Corbato
1990年图灵奖


C. Thacker
2009年图灵奖


Ole-Johan Dahl
2001年图灵奖


Alan Kay
2003年图灵奖


Barbara Liskov
2008年图灵奖


Robert W. Floyd
1978年图灵奖


Leslie Lamport
2013年图灵奖


R. Kahn & V. Cerf
2004年图灵奖


D. Engelbart
1997年图灵奖


Tim Berners-Lee
2016年图灵奖

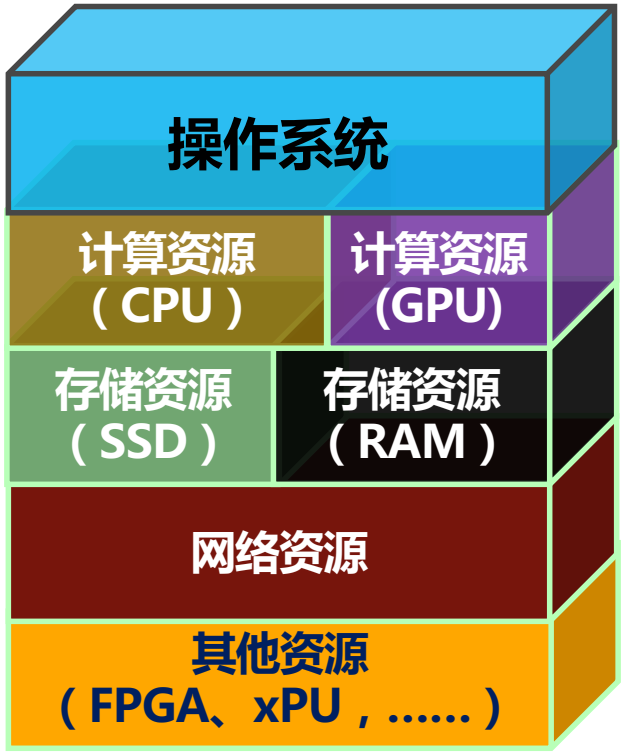
？

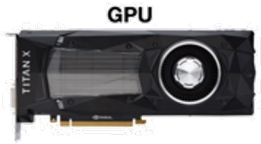
.....

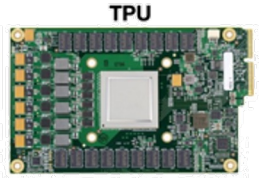
每次操作系统换代时，均有重要基础创新的支撑

操作系统发展的主要驱动力

- 追求更高效地发挥硬件资源所提供的计算能力
  - 通过软件技术形成灵活、高效、可信、统一的虚拟资源

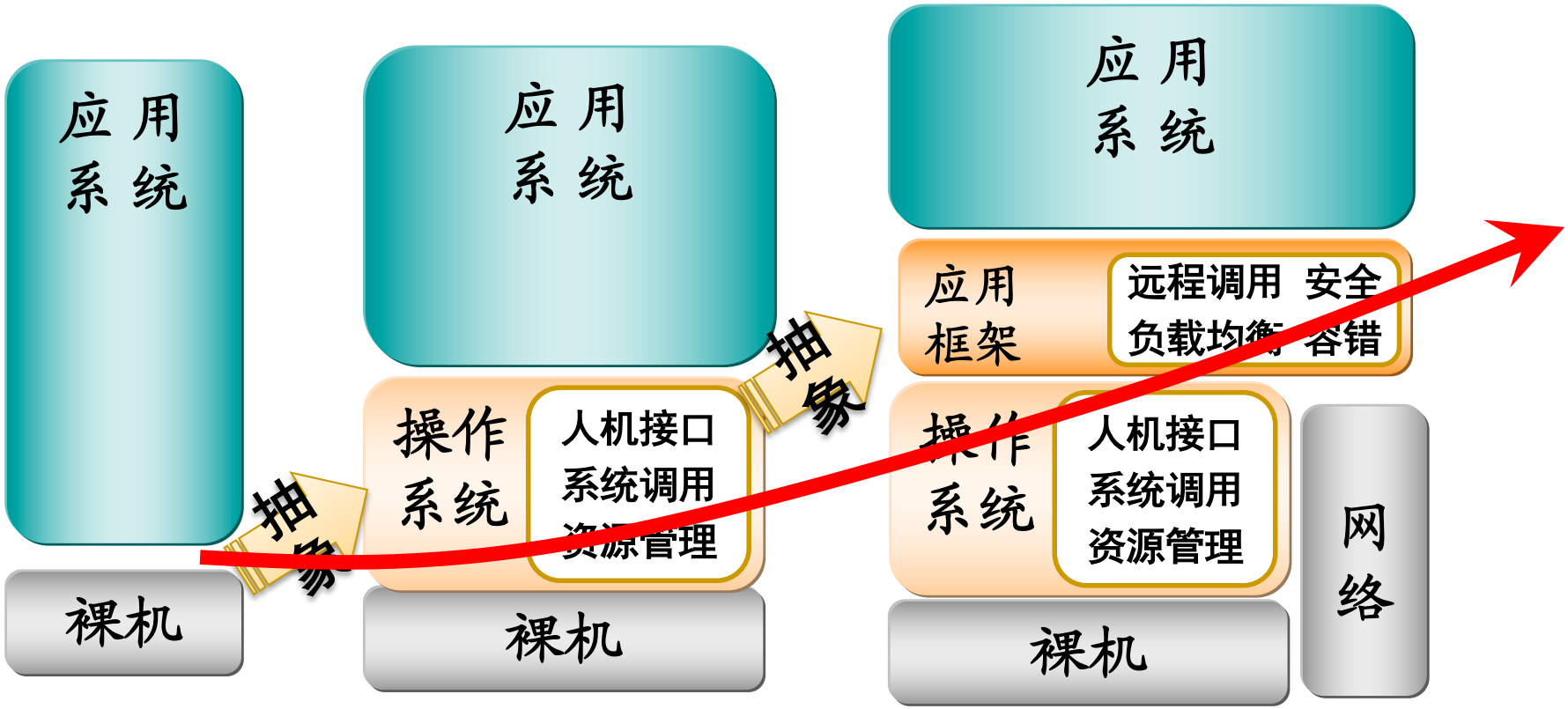






操作系统发展的主要驱动力

- **尽可能提高软件开发和运行的效率和质量**
  - 凝练共性并复用：应用程序的共性沉淀为操作系统的一部分
  - 开发工具不断被融入运行平台，展现了**开发运行平台的合一趋势**



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计算机、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

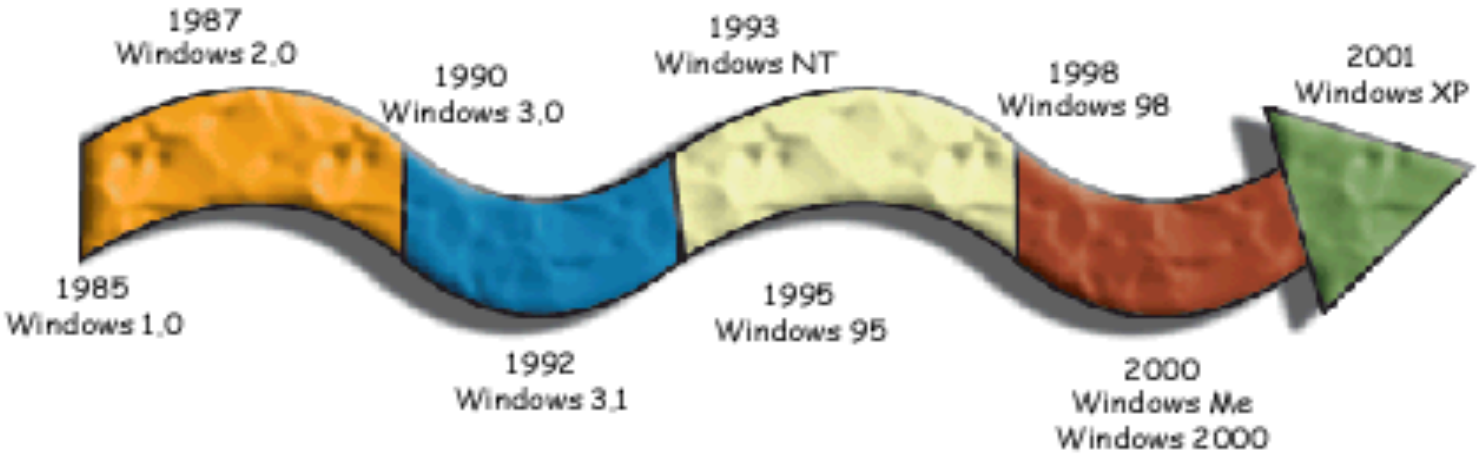
DOS


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.jpx EJBTest.jpx.local
EJBTest.jpx.local~ EJBTest.jpx~
HelloSessionBean.ejbgrpx MyBean.ejbgrpx
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>

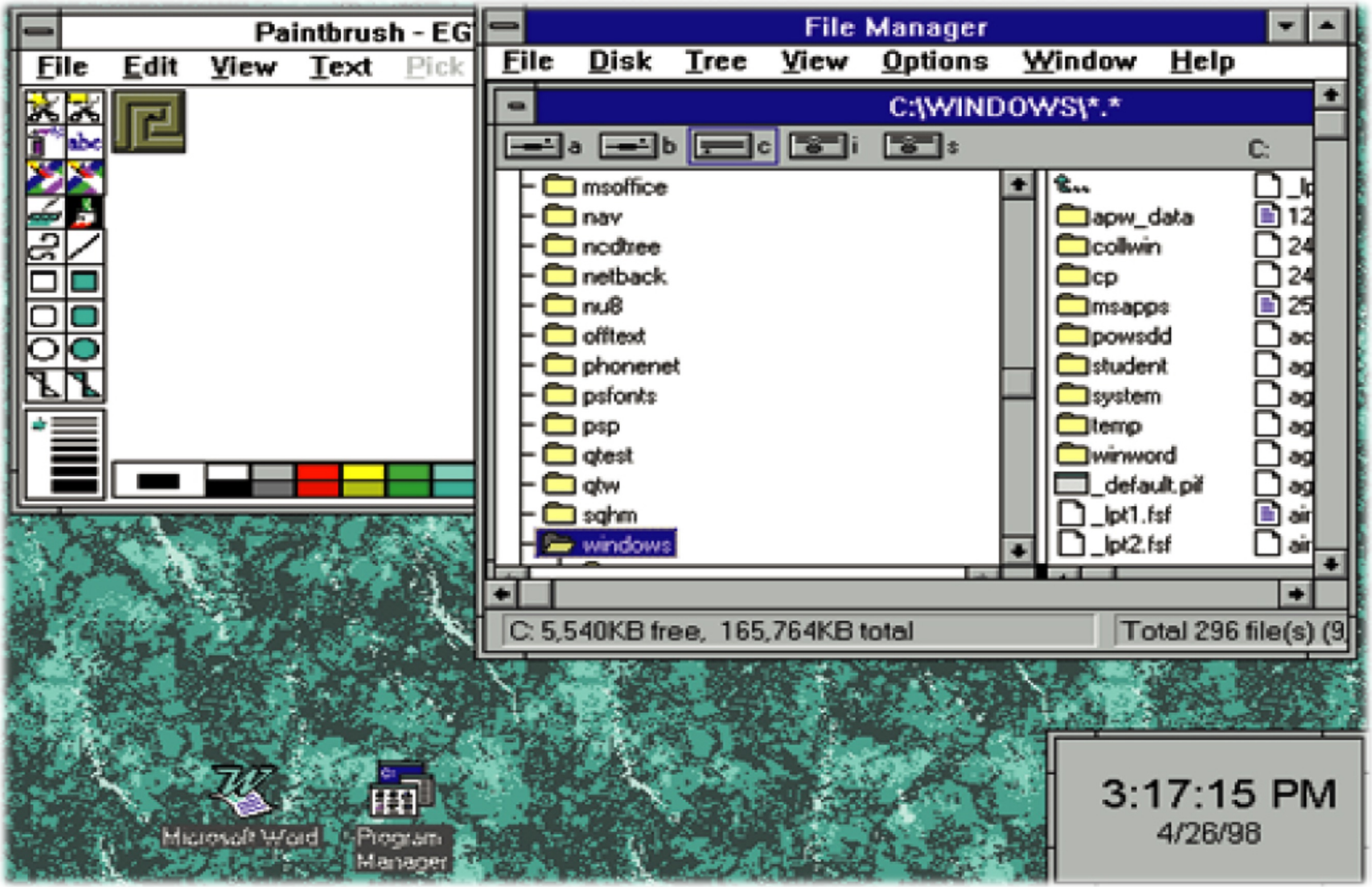
Windows

- 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

最大的特点就是“好用”“傻瓜”



Windows



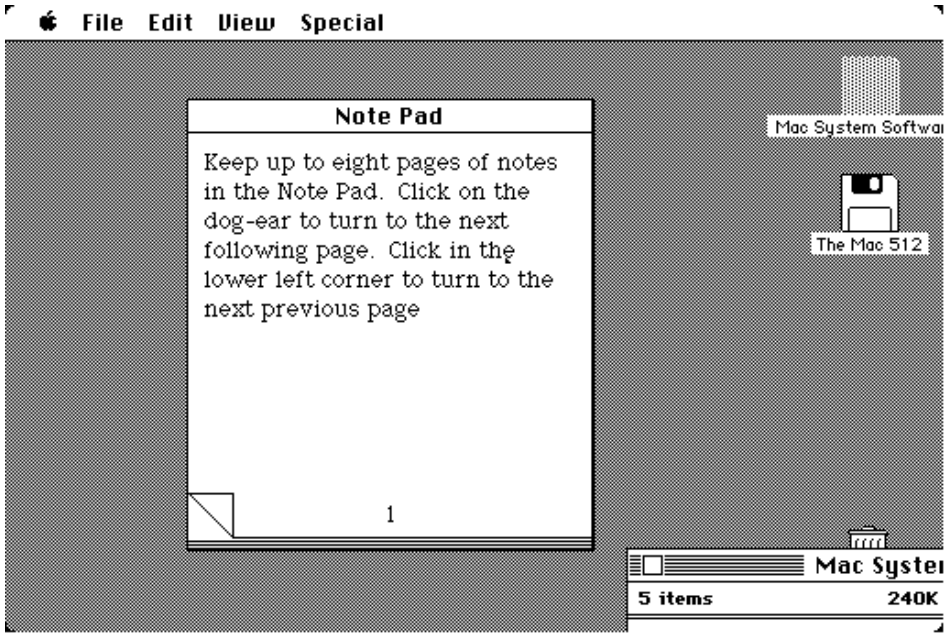
Windows 3.1

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


File Edit View Special
Note Pad
Keep up to eight pages of notes in the Note Pad. Click on the dog-ear to turn to the next following page. Click in the lower left corner to turn to the next previous page
1
Mac System Softwa
The Mac 512
Mac Syste
5 items
240K

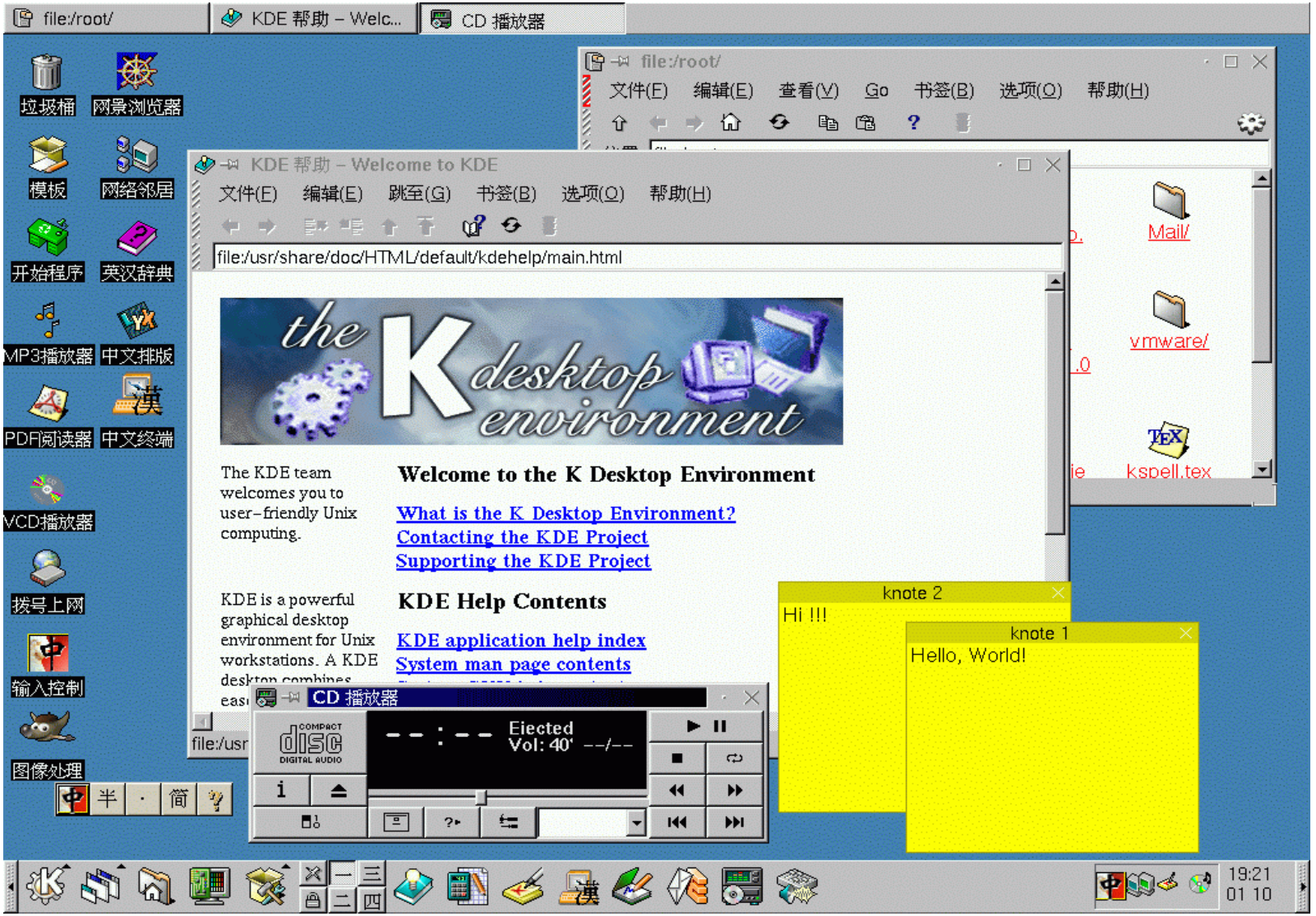

PCPOP.COM 电脑时尚
Mac OS X
PCPOP.COM 电脑时尚


而这种效果贯彻到了整个系统

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


file:/root/
KDE 帮助 – Welc...
CD 播放器
垃圾桶
网景浏览器
模板
网络邻居
开始程序
英汉辞典
MP3播放器
中文排版
PDF阅读器
中文终端
VCD播放器
拨号上网
输入控制
图像处理
KDE 帮助 – Welcome to KDE
文件(F) 编辑(E) 跳至(G) 书签(B) 选项(O) 帮助(H)
file:/usr/share/doc/HTML/default/kdehelp/main.html
the K desktop environment
The KDE team welcomes you to user-friendly Unix computing.
Welcome to the K Desktop Environment
What is the K Desktop Environment?
Contacting the KDE Project
Supporting the KDE Project
KDE is a powerful graphical desktop environment for Unix workstations. A KDE
KDE Help Contents
KDE application help index
System man page contents
file:/root/
文件(F) 编辑(E) 查看(V) Go 书签(B) 选项(O) 帮助(H)
Mail/
vmware/
kspell.tex
knote 2
Hi !!!
knote 1
Hello, World!
CD 播放器
Eiected
Vol: 40'
19:21
01 10

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

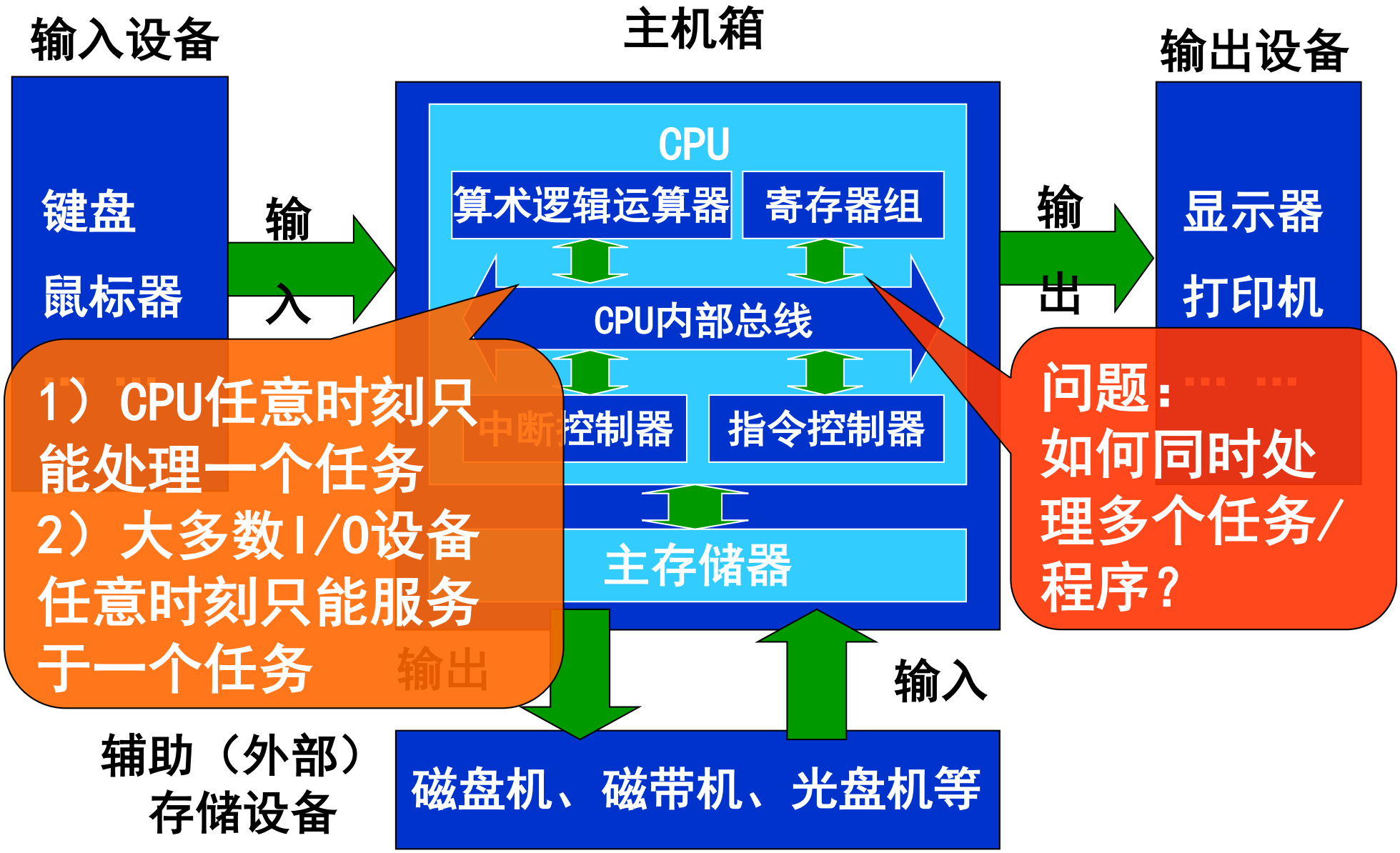
Android是一种以Linux为基础的开放源代码操作系统。2003年投入研发，2005年被Google收购。由于其开放平台的特性，被多家手机制造厂商如三星、摩托罗拉、LG采用。

- 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

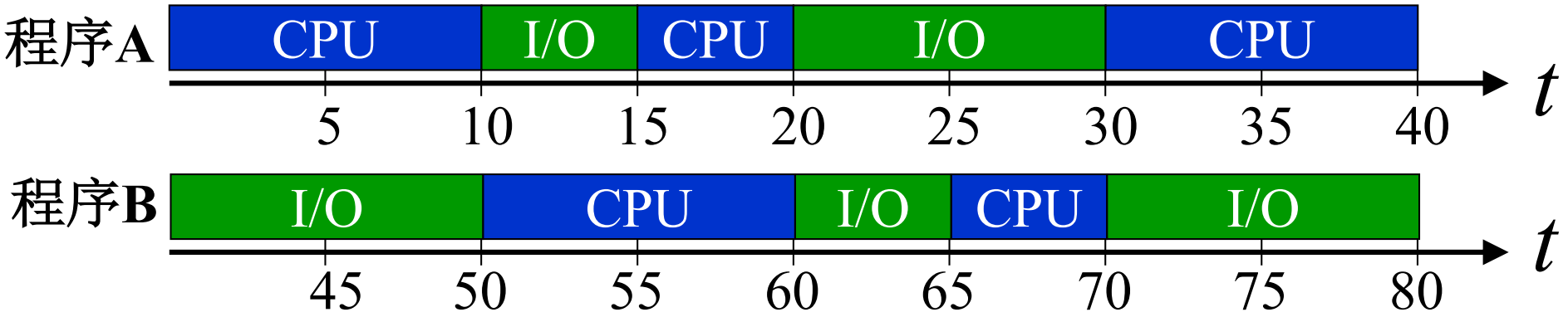
为什么要操作系统？


输入设备
如何管理种类繁多的外设？
主机箱
如何同时处理多个任务？
鼠标器
打印机
设备管理
进程管理
用户界面
内存管理
文件管理
操作系统
Operating System
输出
输入
磁带
方便地
间的数据交换？
Windows Mess...
文件(F) 操作(A) 工具(T) 帮助(H)
PKUAS Features in JonAS 5(联机)
添加联系人
计算器
编辑(E) 查看(V) 帮助(H)
Backspace CE C
正在播放
后退 前进 停止 刷新 主页
准备就绪
计算机概论
未命名 - 画图
文件(F) 编辑(E) 查看(V) 图像(I) 颜色(C) 帮助(H)
颜色选择
编辑工具箱
要获得帮助，请在“帮助”菜单中，单击“帮助”
Google 桌面 - Microsoft Internet Explorer
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
http://127.0.0.1:4664/?s=Z6SOETQg5LUMPfP-HuOCPQGTFDw
Google 桌面
网页 图片 新闻 论坛 网页目录 桌面 更多»
搜索桌面 搜索网页
桌面使用偏好
浏览时间线
搜索您的计算机。
完毕
Internet

为什么需要进程管理


输入设备
主机箱
输出设备
键盘
鼠标器
输入
CPU
算术逻辑运算器
寄存器组
CPU内部总线
控制器
指令控制器
主存储器
输出
显示器
打印机
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？
输入
输出
辅助（外部）存储设备
磁盘机、磁带机、光盘机等

为什么需要进程管理

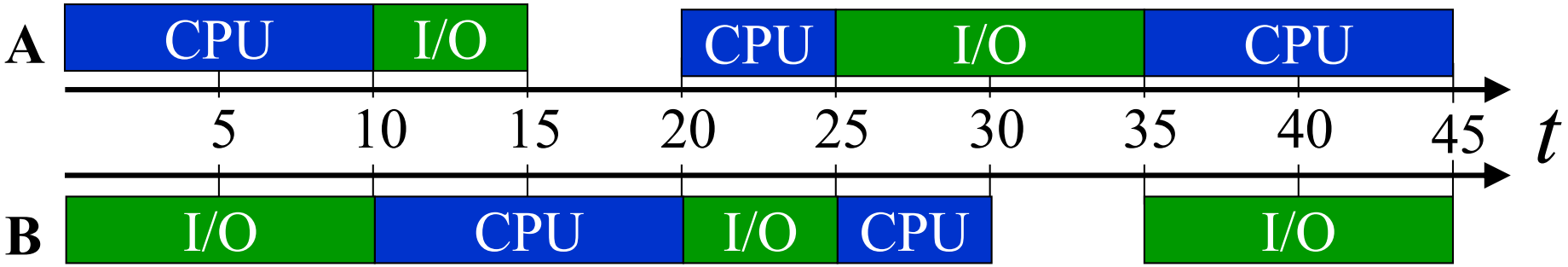


❑程序的顺序运行

- A先运行，B再运行
- CPU利用率= 40/80 = 50%
- I/O利用率= 40/80 = 50%

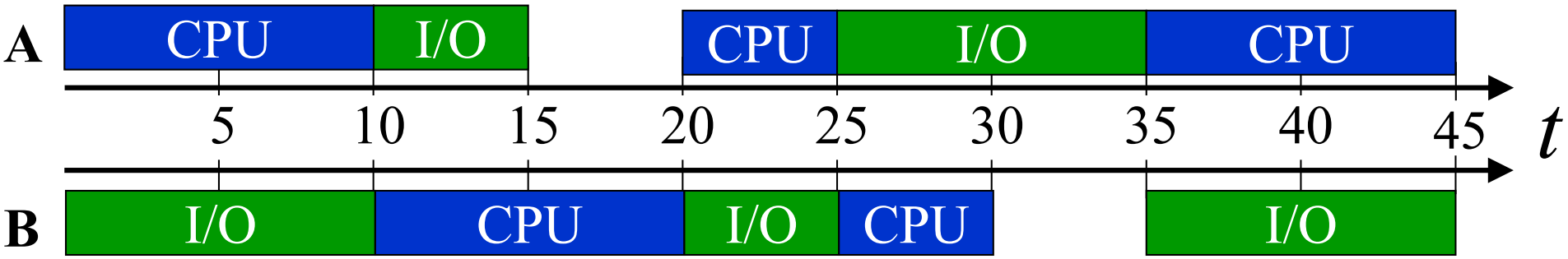
❑程序的并发运行

- 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



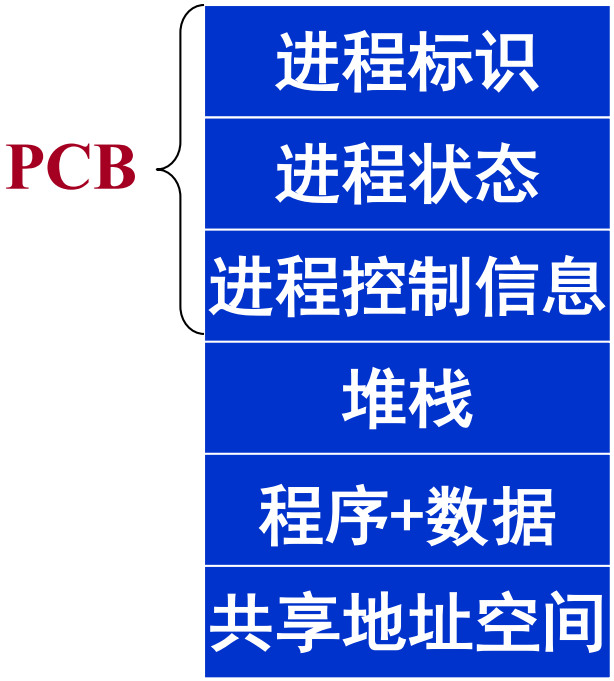
进程：使能程序并发运行

❑ 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

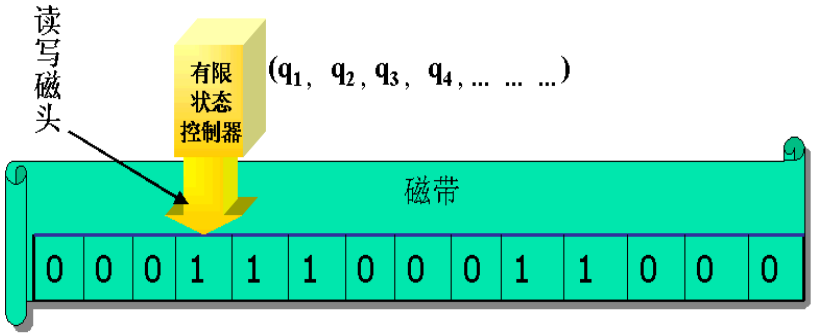
- ➢ 用户程序（代码）
- ➢ 用户数据
- ➢ 堆栈（用于过程调用和参数传递）
- ➢ 进程控制块**PCB (**进程属性**)**



程序与进程

❑ 程序与进程的区别

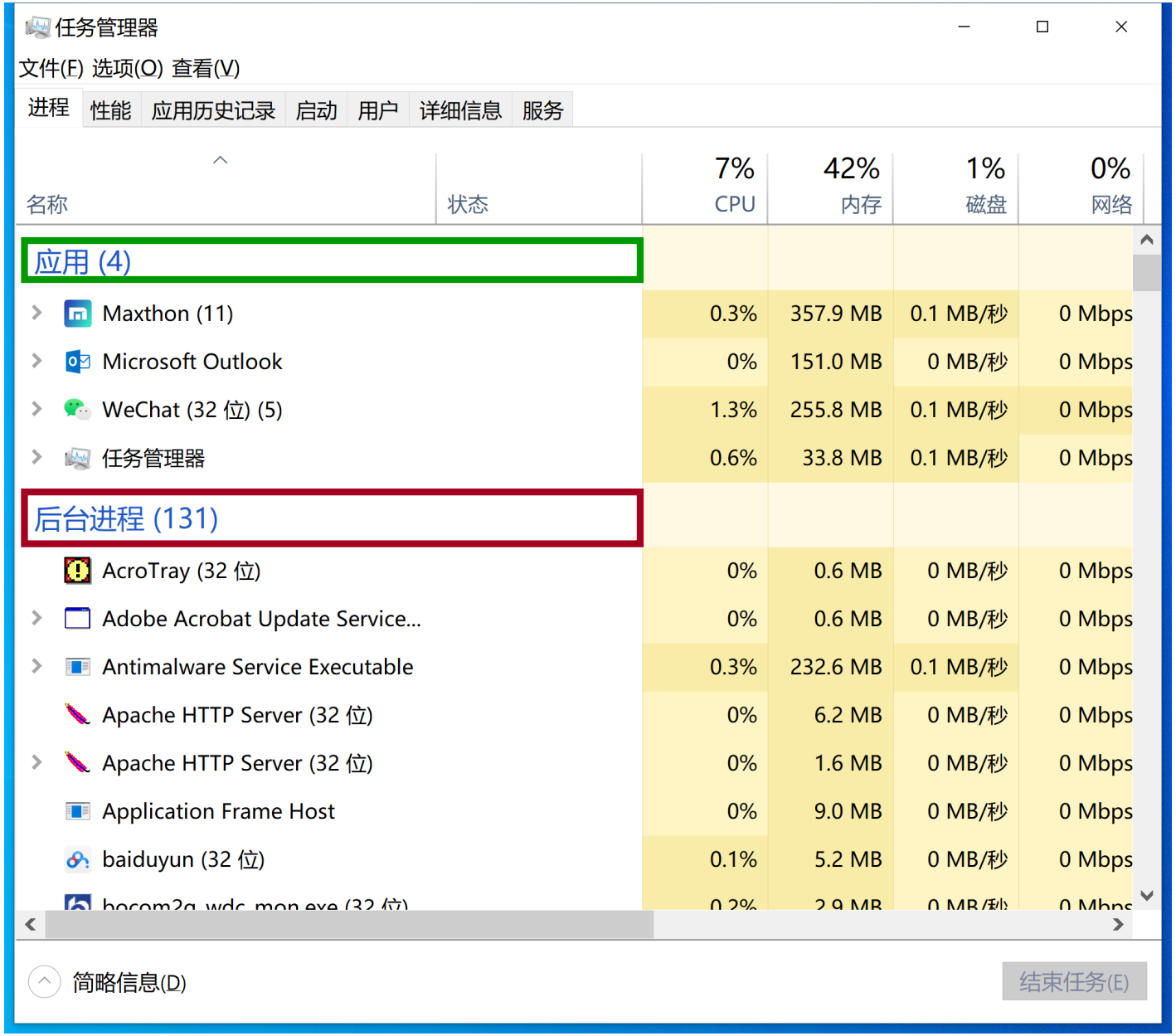
- 进程更能真实地描述并发，而程序不能
- 进程是由程序和数据两部分组成的
- 程序是静态的，进程是动态的，描述程序执行时动态特征
- 进程有生命周期，有诞生有消亡，短暂的；而程序是相对长久的
- 一个程序可对应多个进程，反之亦然
- 进程具有创建其他进程的功能，而程序没有



在前面的约定下，计算$f(x) = 2x$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

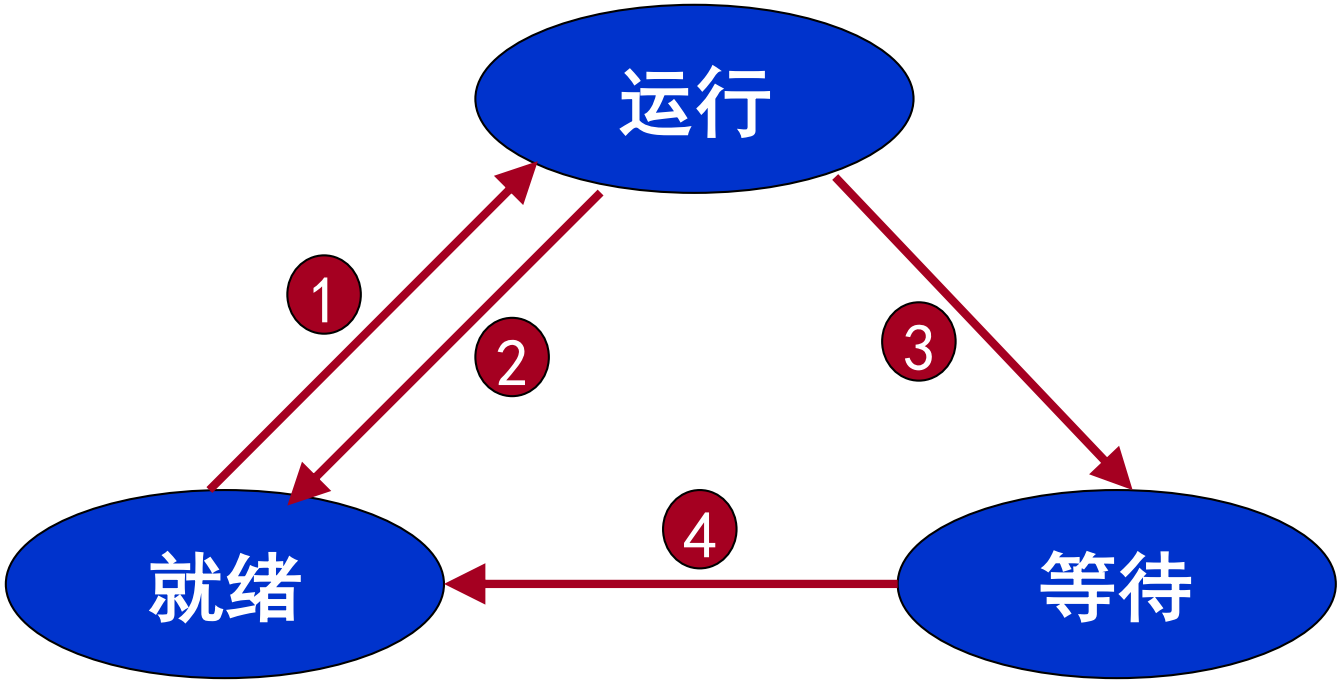


进程的分类：系统进程和用户进程（前者优于后者）

进程的状态

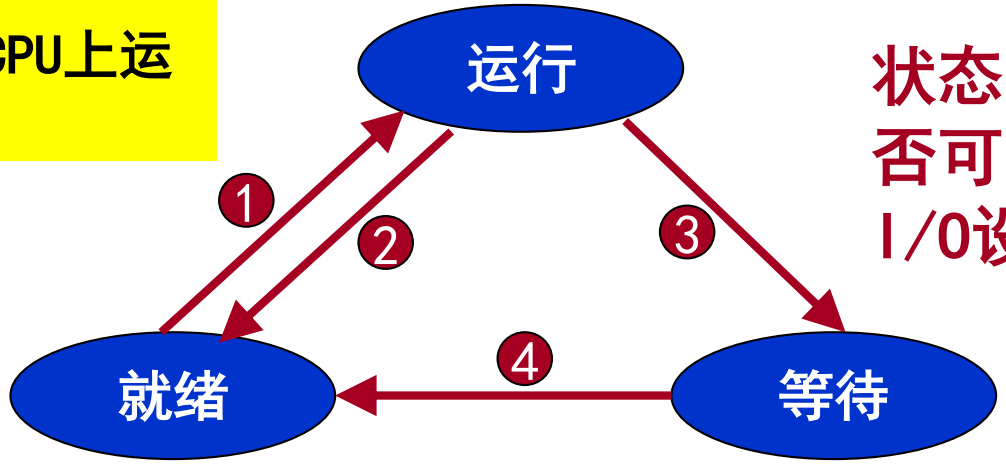
❑ 进程的三种基本状态

- ➢ 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

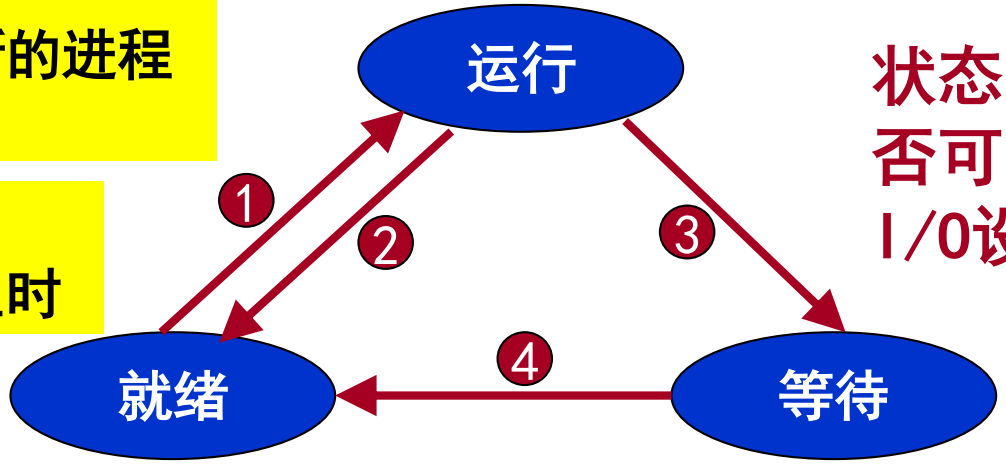
就绪态（Ready）：
一个进程已经具备运行条件，但由于无CPU暂时不能运行的状态（当调度给CPU时，立即可以运行）

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



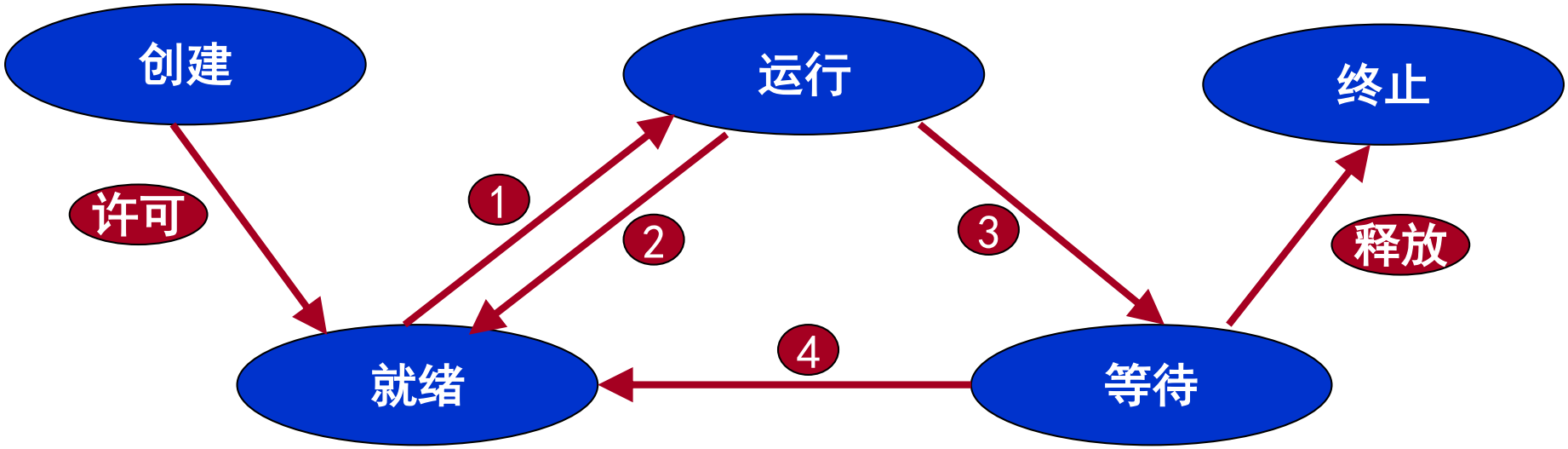
状态表明进程是否可以使用CPU和I/O设备

（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	Memory management	File management
Registers	Pointer to text segment	Root directory
Program counter	Pointer to data segment	Working directory
Program status word	Pointer to stack segment	File descriptors
Stack pointer		User ID
Process state		Group ID
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		

线程：
轻量级进程

进程标识

进程状态

进程控制信息

堆栈

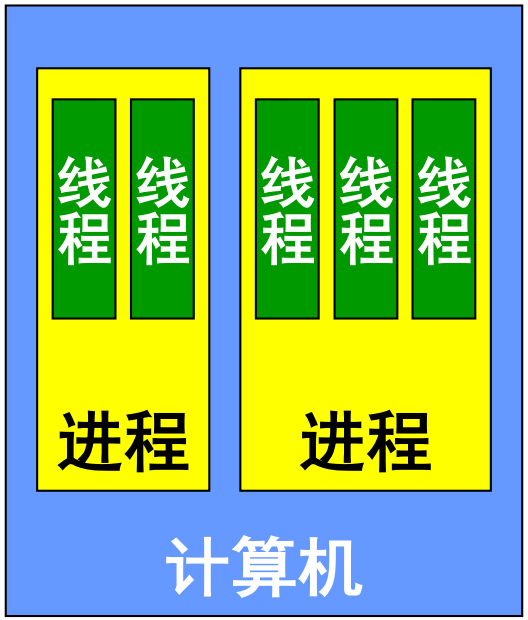
程序+数据

共享地址空间

进程与线程

Thread

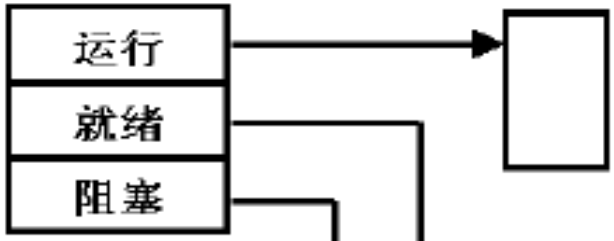
- 是进程中的一个实体，是被CPU执行任务的实体
- 线程自己不拥有系统资源，只拥有在执行中必不可少的资源(如程序计数器、一组寄存器和栈等)，但是它可以和同属一个进程的其他线程共享进程所拥有的全部资源
- 更适合于多任务调度
  - 便于将进程分割成多个并发单元
  - 线程附带的资源少因而创建更快



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？

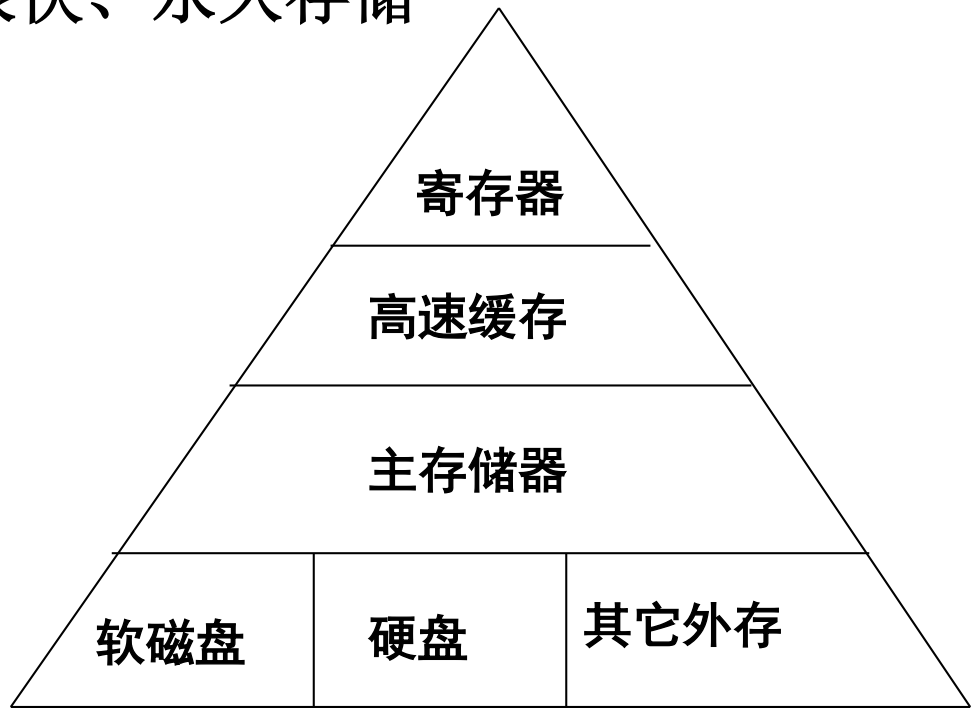
4）如何避免多个进程之间产生死锁？

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

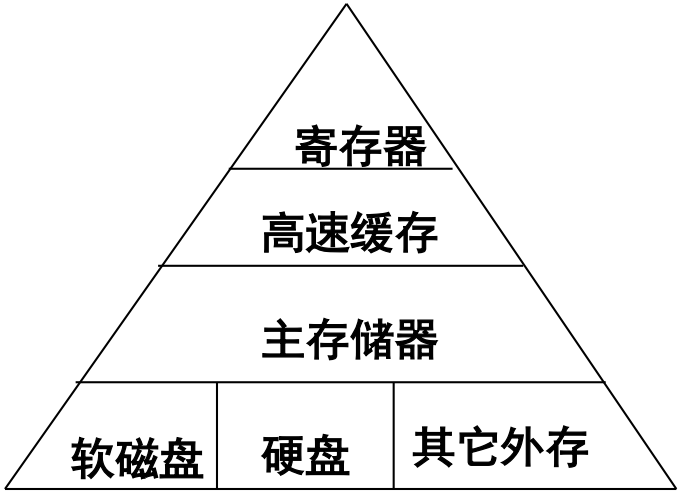
- 帕金森定律：“内存多大，程序多长”
  - 程序大小的增长速度比内存容量的增长快
- 程序员的梦想
  - 内存容量无限大、速度无限快、永久存储
- **操作系统存储管理**
  - **充分利用内存，为程序并发执行提供存储基础**
  - **尽可能方便用户使用，如自动装入用户程序**
  - **解决程序空间比实际内存空间大的问题**



存储管理

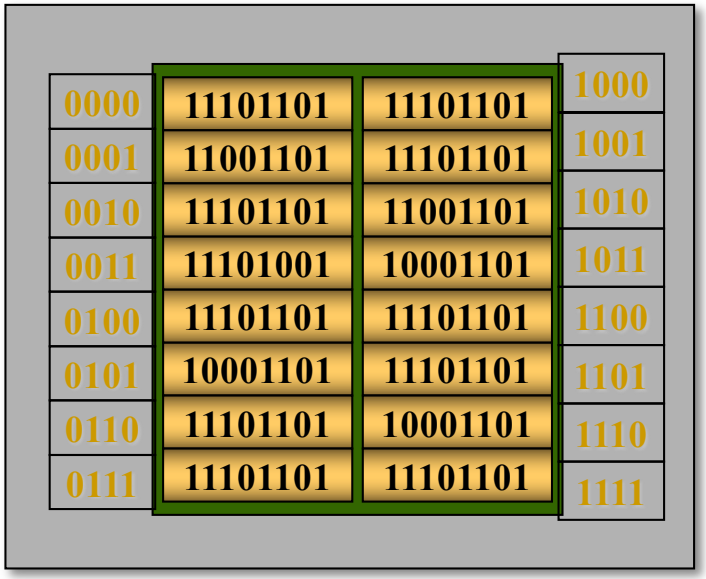
❑ 操作系统存储管理

- ➢ 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



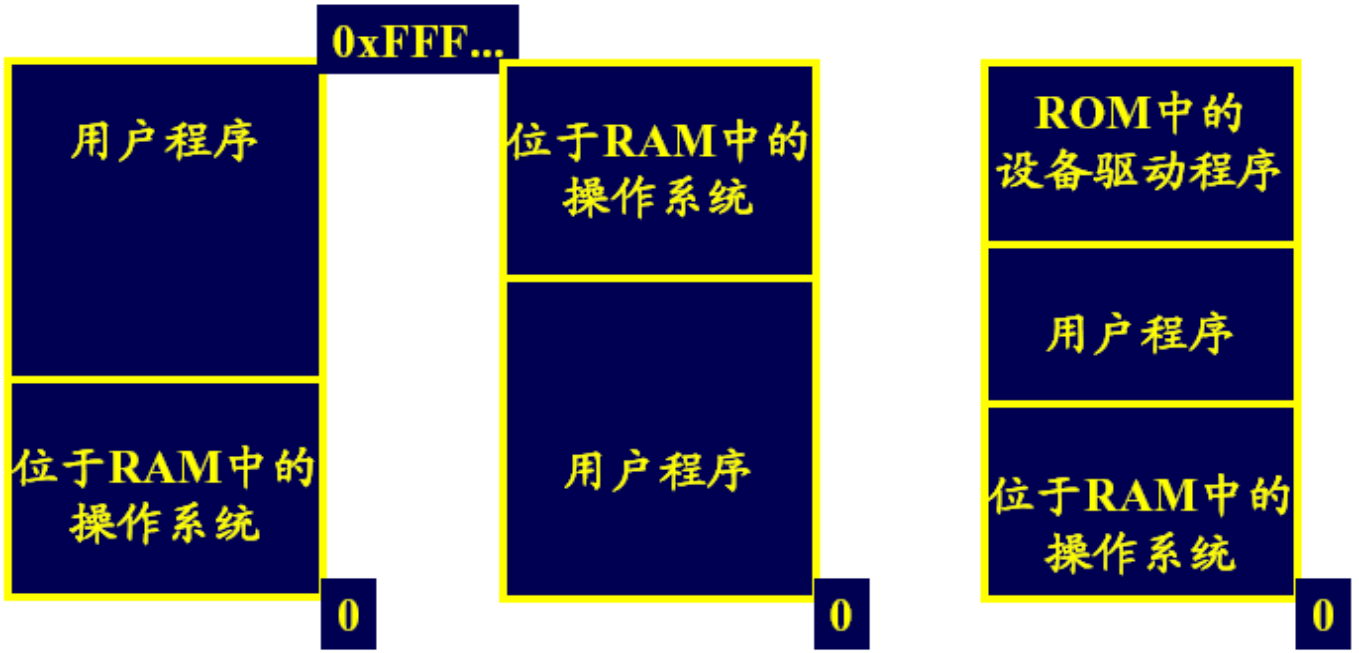
❑ 存储管理技术分类

- ➢ 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



单一用户（连续区）存储管理方案

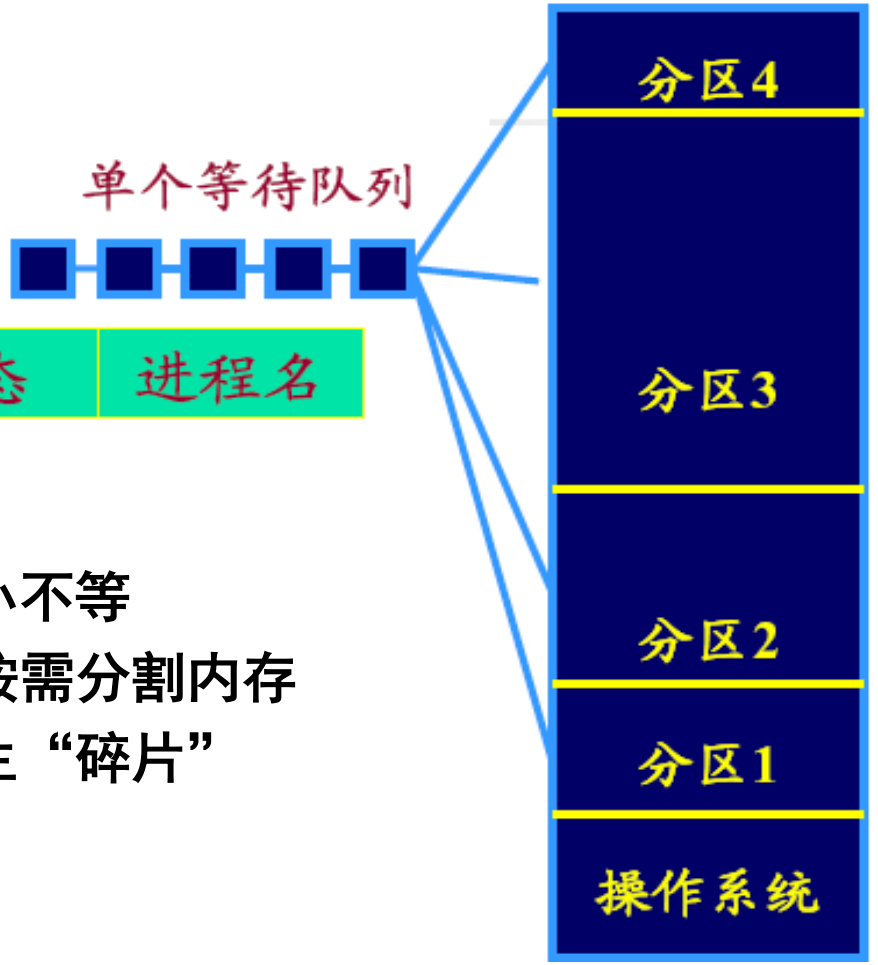
- 单用户系统在一段时间内，只有一个进程在内存
  - 内存分配管理十分简单，内存利用率低



- 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



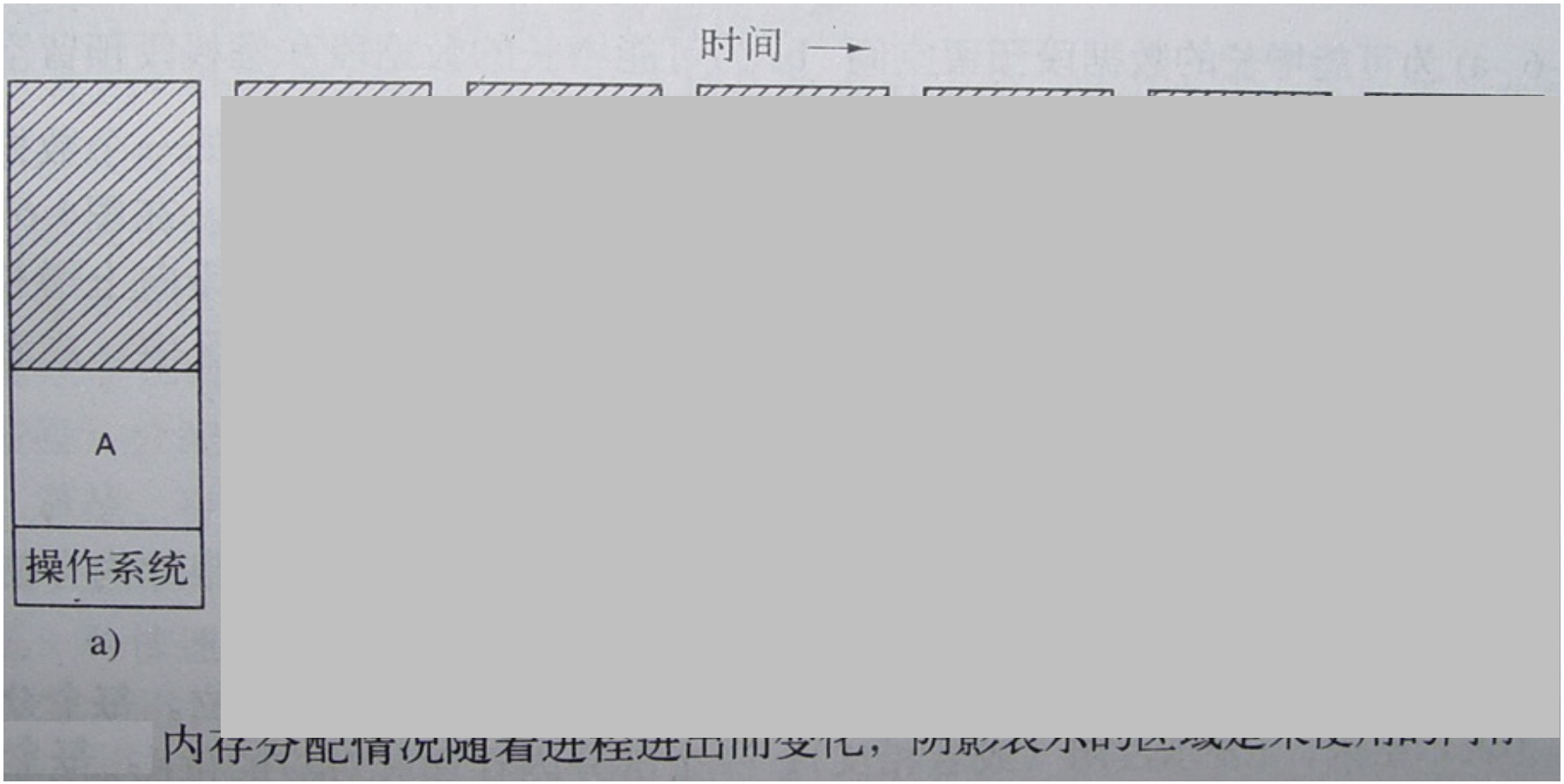
分区号	起始地址	长度	状态	进程名

- 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

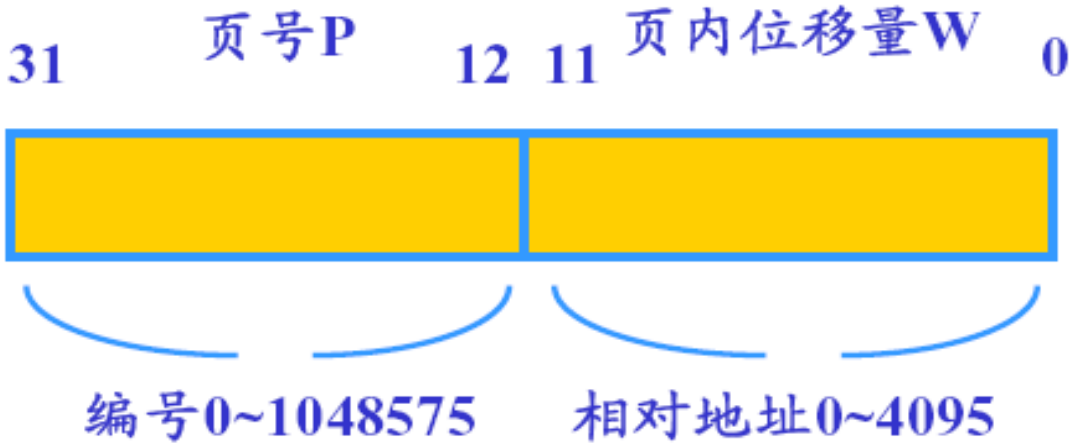
- ➢ 操作系统自动分割程序
- ➢ 对程序员而言，得到了比实际内存容量大得多的内存空间
  - 如现在的Windows，整个硬盘都可以做成虚拟主存
  - 在运行游戏或图像处理软件时，硬盘灯狂闪就是操作系统在交换数据

- ➢ 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

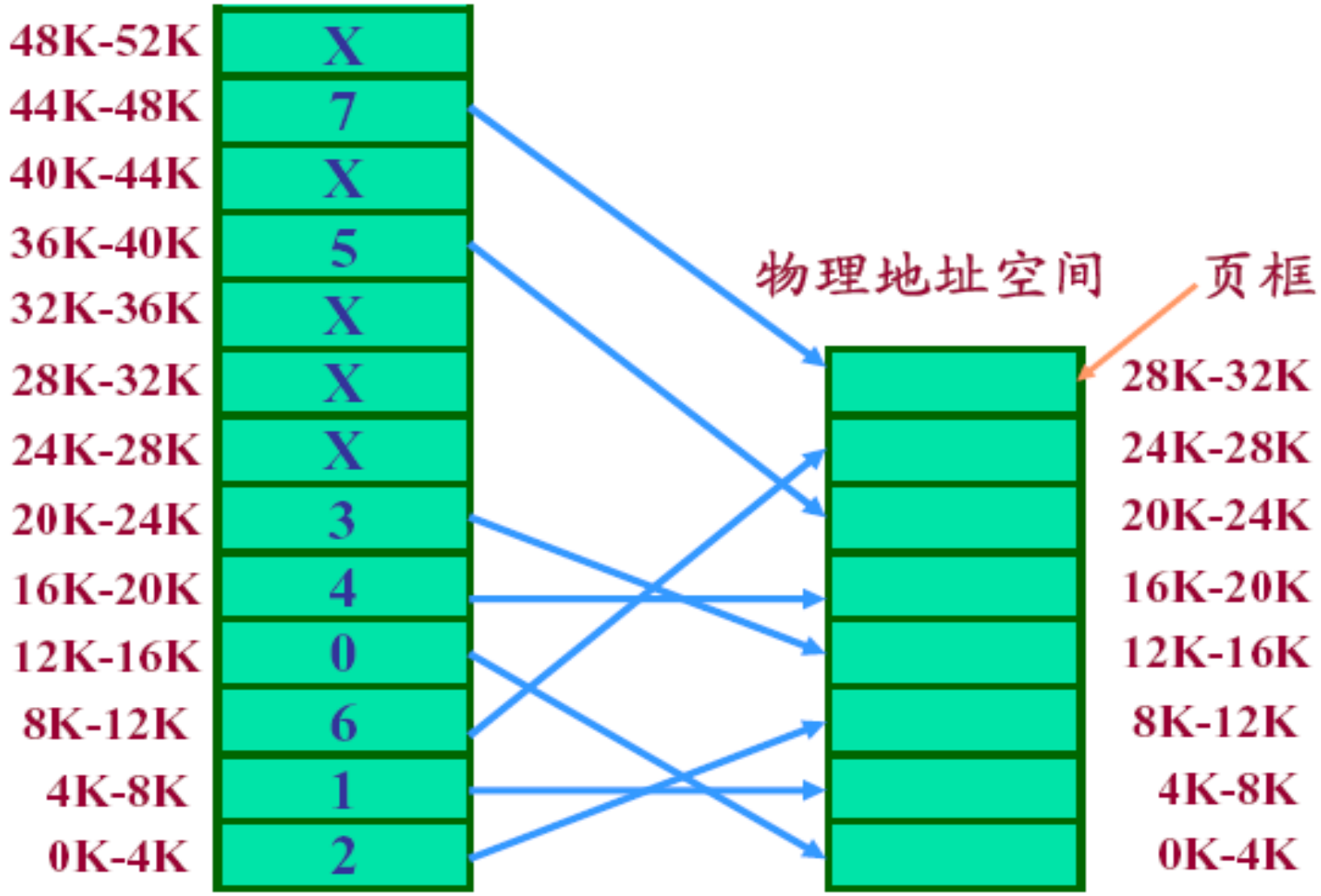
❑基本思想（工作原理）

- ➢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



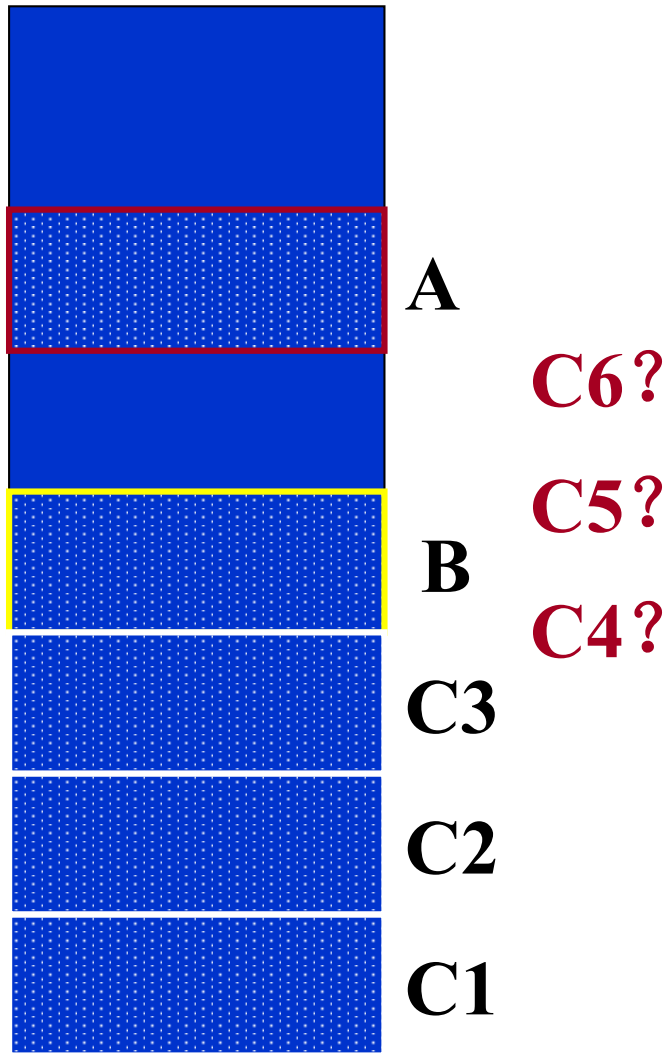
段式存储管理方案

❑ 页式存储管理方案的不足

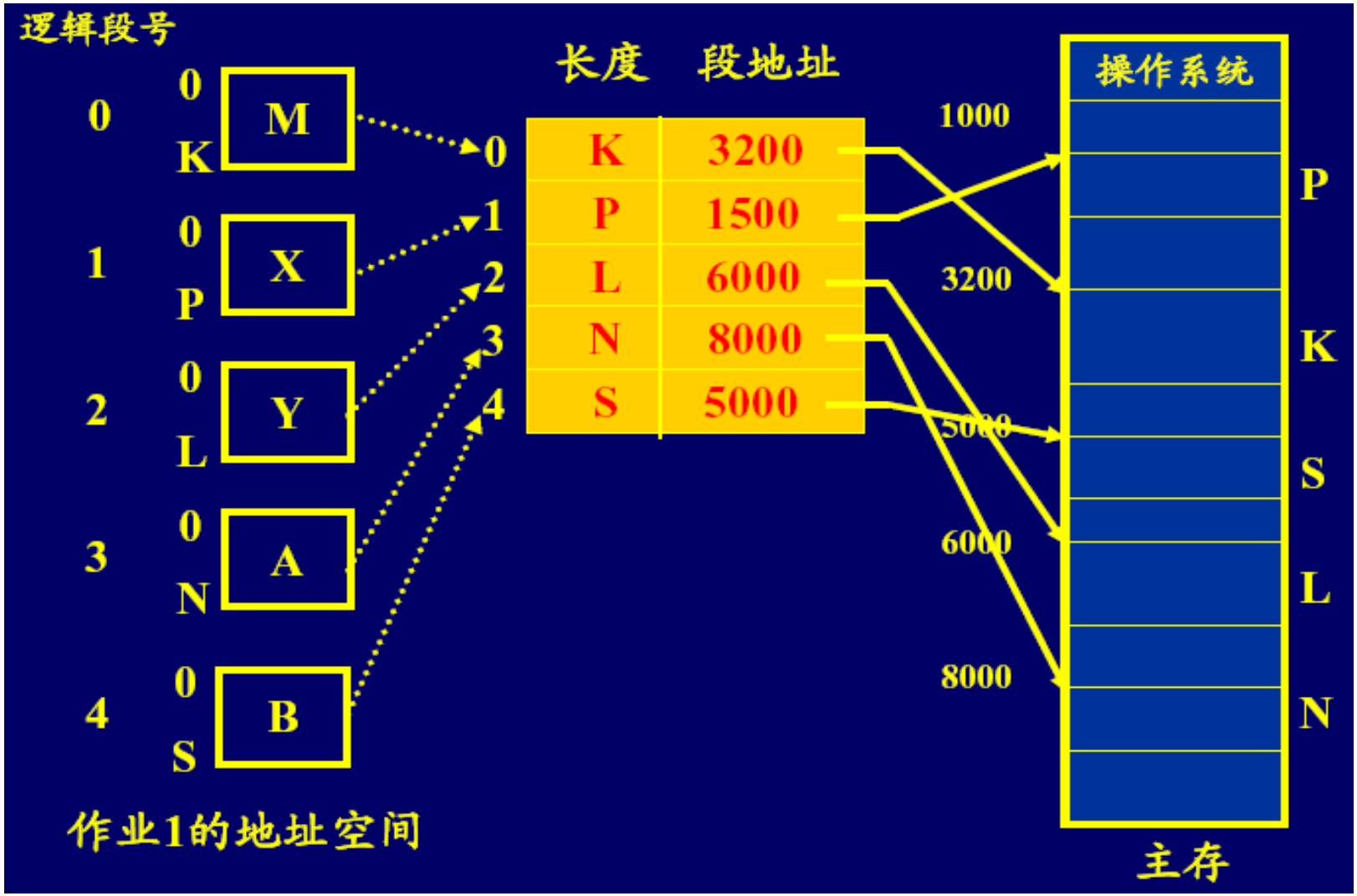
- ➢ 页的大小固定，不适合大小动态变化的数据块

❑ 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


逻辑段号
0
1
2
3
4
0
K
M
0
P
X
0
L
Y
0
N
A
0
S
B
作业1的地址空间
长度
段地址
K
3200
P
1500
L
6000
N
8000
S
5000
0
1
2
3
4
1000
3200
5000
6000
8000
操作系统
P
K
S
L
N
主存

段页式存储管理方案

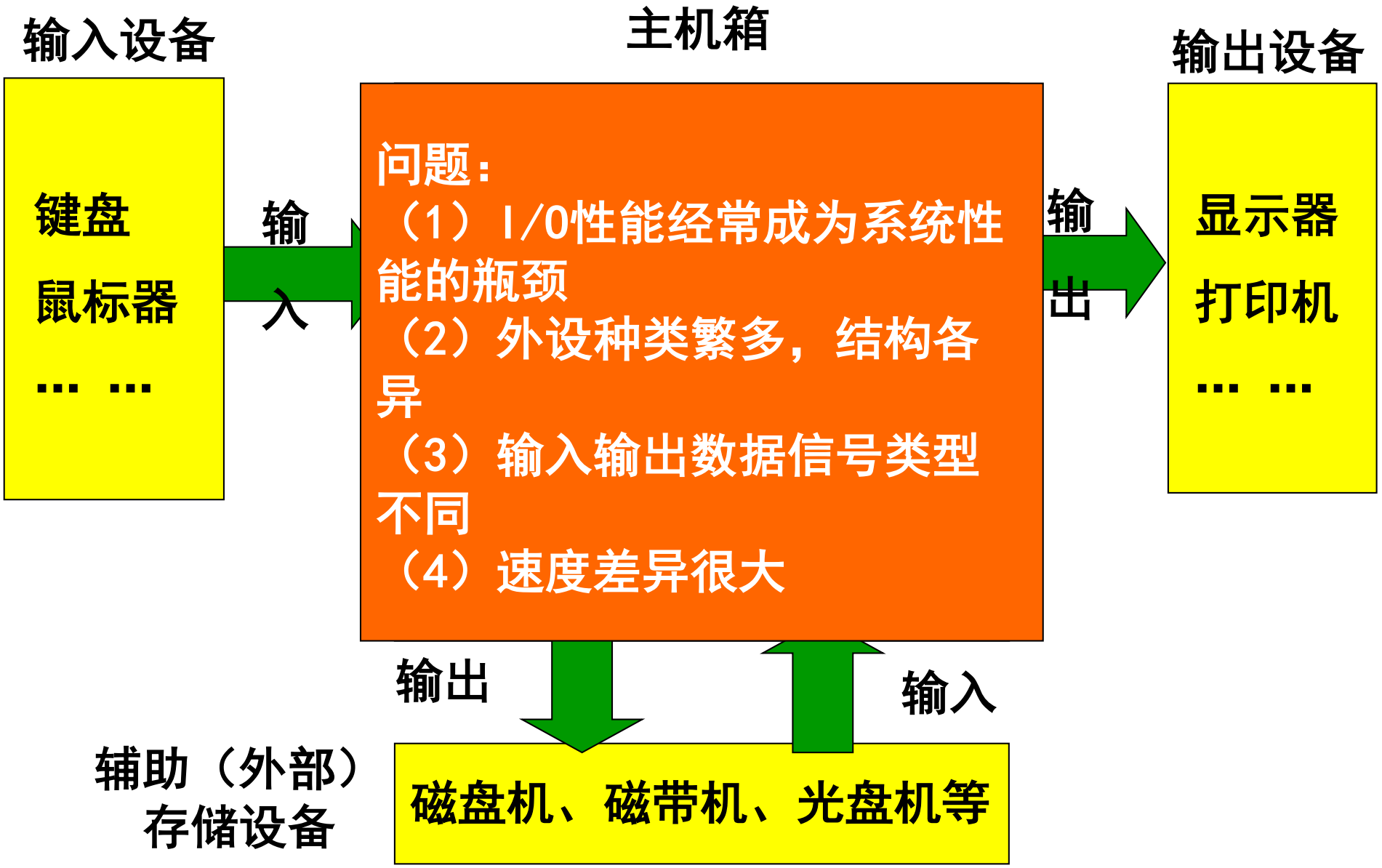
- 分段较大时，并不一定要全部放在主存
  - 也可能整个主存都放不下
- 自然地，对段进行分页
  - 对用户程序分段
  - 对内存分页

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


输入设备
主机箱
输出设备
键盘
鼠标器
… …
输入
问题：
（1）I/O性能经常成为系统性能的瓶颈
（2）外设种类繁多，结构各异
（3）输入输出数据信号类型不同
（4）速度差异很大
输出
显示器
打印机
… …
输出
输入
辅助（外部）存储设备
磁盘机、磁带机、光盘机等

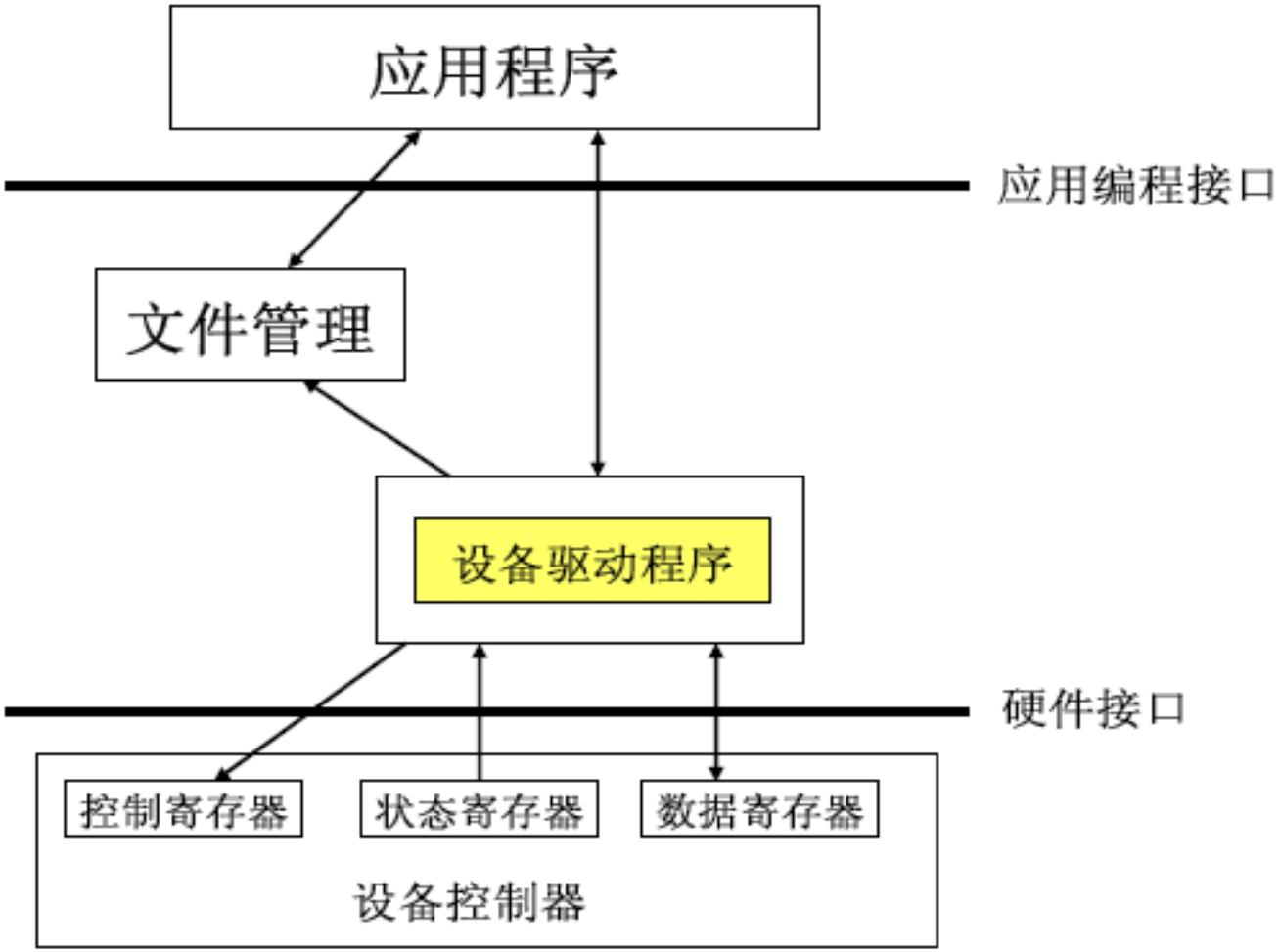
为什么要设备管理

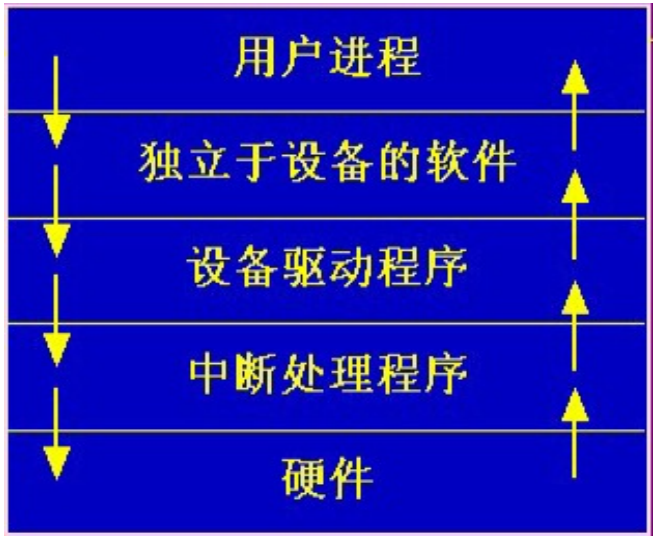
- 完成用户的I/O请求
  - 按照用户的请求，控制设备的各种操作，完成I/O设备与内存之间的数据交换
  - 包括设备分配与回收；设备驱动程序；设备中断处理；缓冲区管理
- 方便的编程接口、设备独立性
  - 逻辑设备与物理设备、屏蔽硬件细节（设备的物理细节，错误处理，不同I/O的差异性）
  - 用户能独立于具体物理设备而方便的使用设备

为什么要设备管理

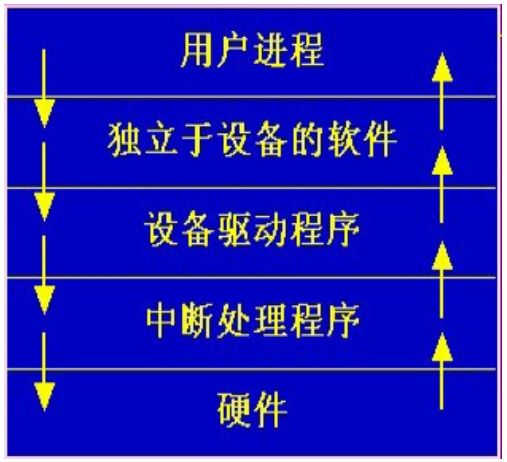
- 完成用户的I/O请求
- 方便的编程接口、设备独立性
- 协调多个进程间的竞争、提高资源利用率
  - 保证在多道程序环境下，当多个进程竞争使用设备时，按一定策略分配和管理各种设备，使系统能有条不紊的工作
  - 充分利用各种技术（通道，中断，缓冲等）提高CPU与设备、设备与设备之间的并行工作能力
- 保护设备传送或管理的数据（安全、无损、保密）

设备管理


应用程序
应用编程接口
文件管理
设备驱动程序
硬件接口
控制寄存器
状态寄存器
数据寄存器
设备控制器


用户进程
独立于设备的软件
设备驱动程序
中断处理程序
硬件

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

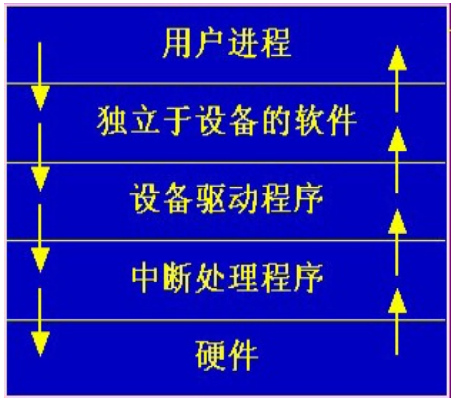
（4）中断处理程序负责I/O完成时，唤醒设备驱动程序进程，进行中断处理

（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



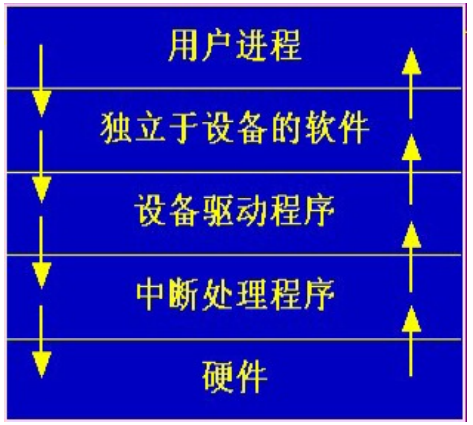
❑ DMA（Direct Memory Access）

- ➢ 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



操作系统的功能

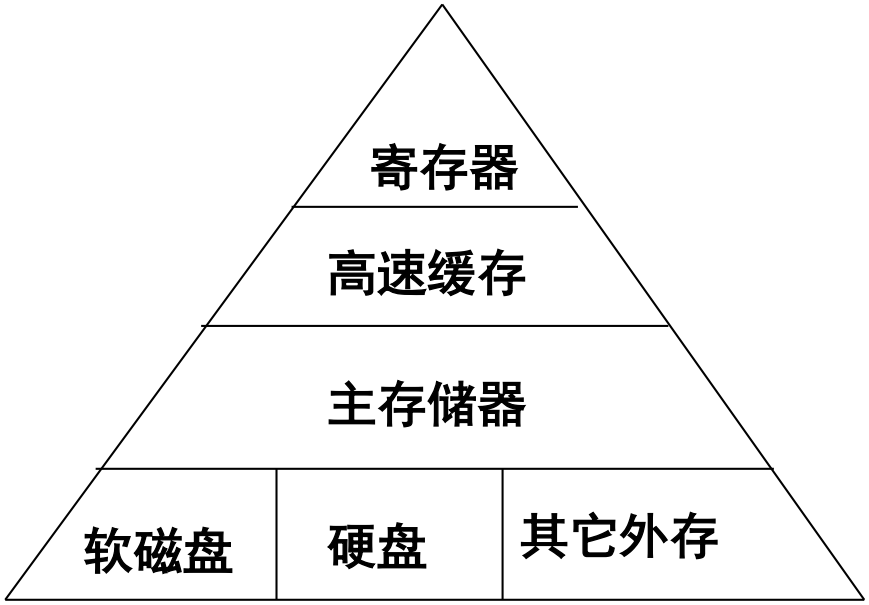
进程管理：
调度CPU和分配系统资源

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

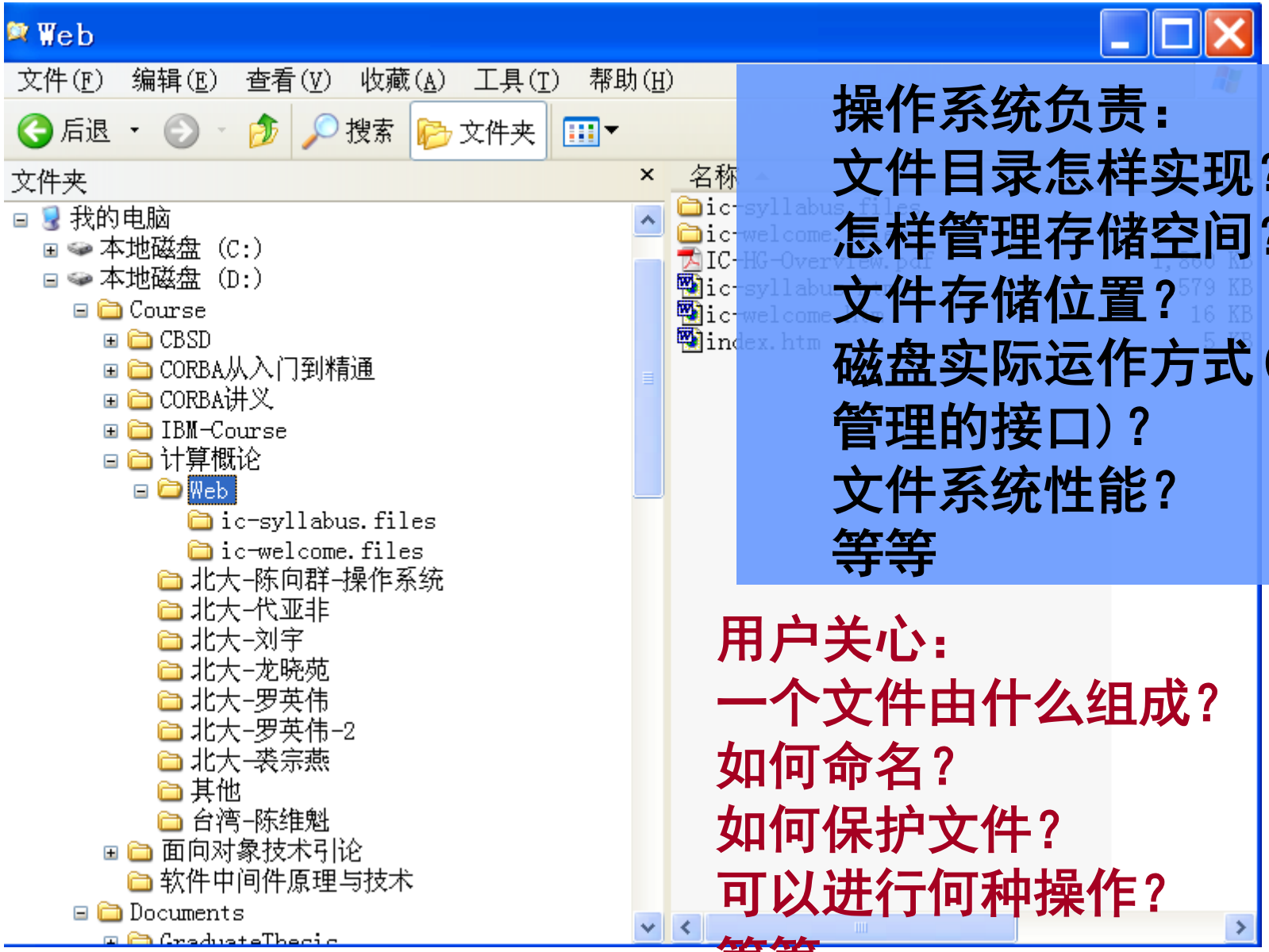
设备管理：
分配和回收外部设备以及控制外部设备按用户程序的要求进行操作

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

更多内容在程序设计中介绍

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

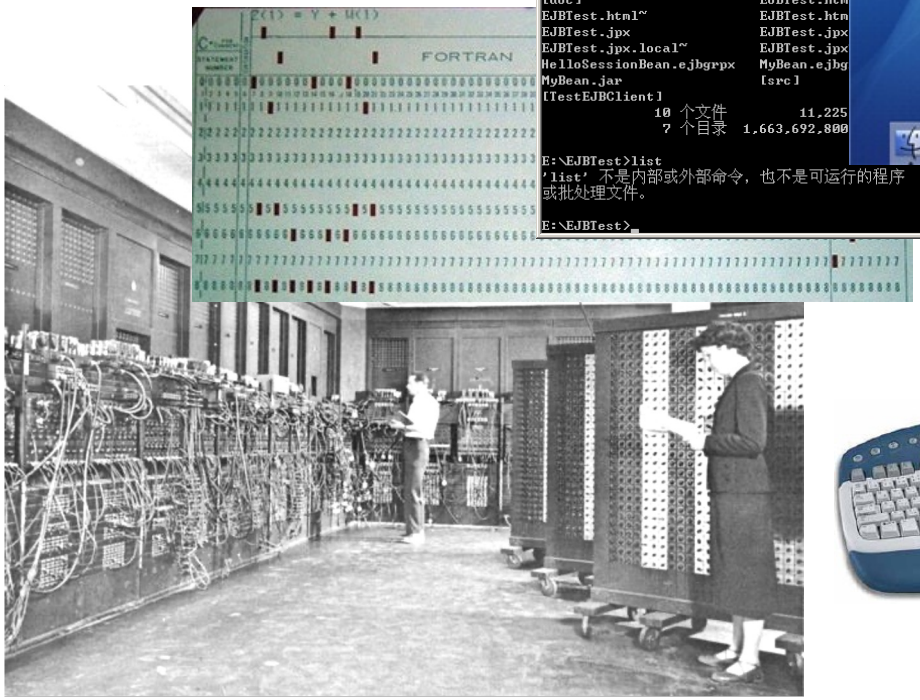
设备管理：
分配和回收外部设备以及控制外部设备按用户程序的要求进行操作

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



用户界面的演变

facebook

Meta

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