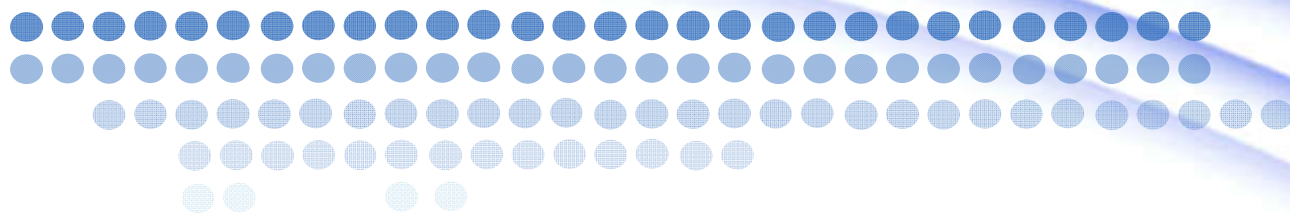


2021级期末考试复习版本



《操作系统原理》

第1章 操作系统概述

教师：邹德清，李珍，苏曙光

华中科技大学网安学院

2023年10月-2024年01月

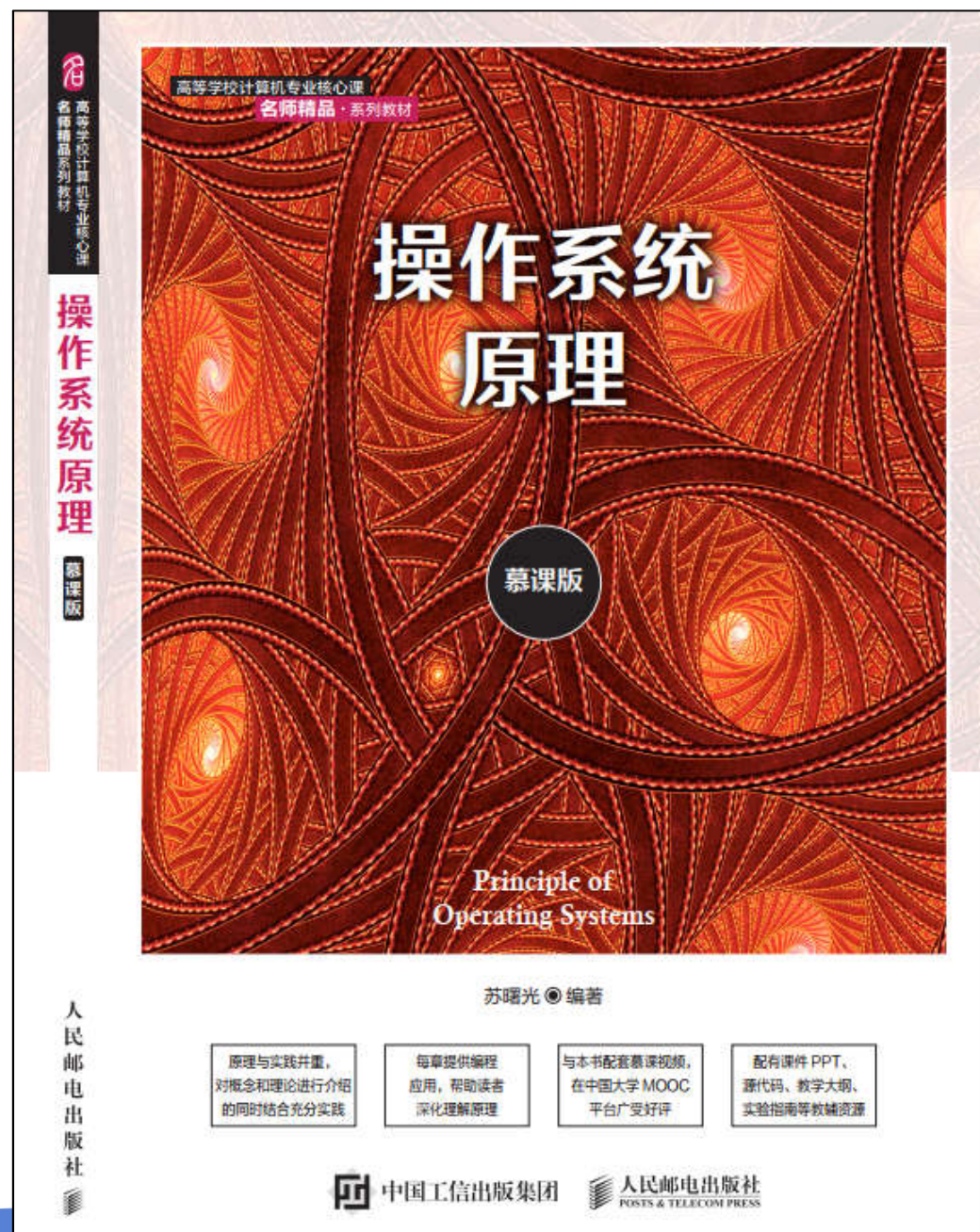
● 苏曙光. 操作系统原理

■ 人民邮电出版社

■ 结构完整

■ 案例丰富

■ 源代码分析



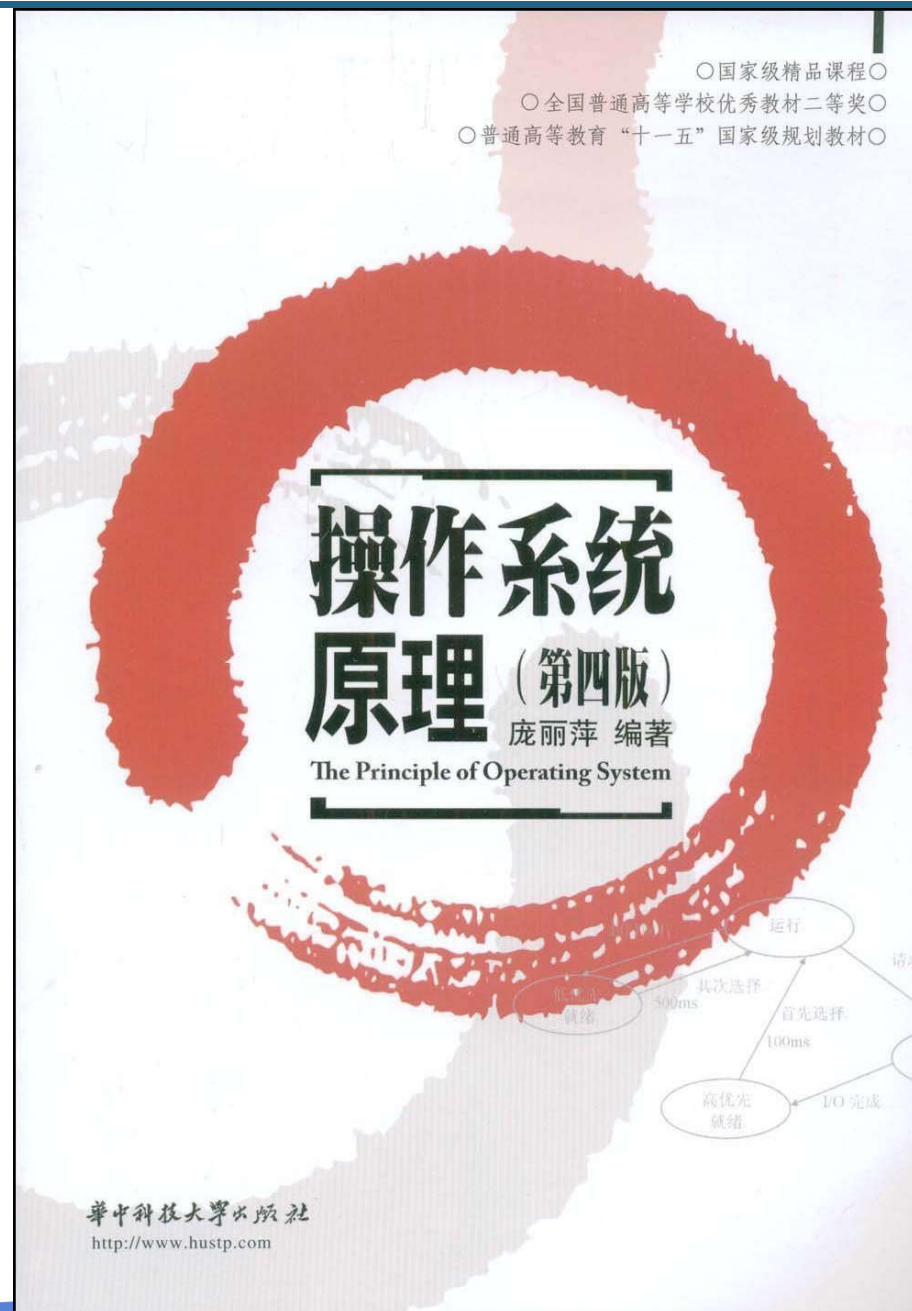
参考书 (1)

● 庞丽萍. 操作系统原理

■ 华中科技大学出版社

■ 结构完整

■ 概念清晰



期末成绩评定

● 理论课

■ 1) 期末考试: 80%

◆ 闭卷

■ 2) 课外作业: 10%

◆ 共四次: 第123章(纸质作业), 第456章(MOOC单元测验), 第7章(MOOC单元测验), 期末测试(MOOC期末考试, 要求线下考试前作为课外作业完成).

■ 3) 考勤: 10%

◆ 点名考勤

◆ 课堂提问

◆ 缺一次扣2分, 最多扣10分

实验课程内容、提交要求和成绩评定(2023.12.18更新)

● 实验课程

- 全课程共4次实验（含华为专家鸿蒙专题，鸿蒙专题不要求提交实验报告）。每次实验有实验任务3-6项，每项实验任务有必做和选做之分。
- （1）现场演示（5%）
 - ◆ 每次实验能当场演示老师预先指定的其中1~3个实验任务。至少成功演示3个，缺一个扣1分。最多得5分】
- （2）实验源代码（40%）
 - ◆ 每个实验任务的源代码功能完整，代码完整可编译，代码注释规范充分。
- （3）实验报告（55%）
 - ◆ 每次实验按群中模板写一份实验报告（除鸿蒙专题），累计应该提交3个实验报告；内容充实；图文并茂；排版规范；无抄袭雷同。

实验课程内容、提交要求和成绩评定(2023.12.18更新)

● 实验课程

■ 实验资料提交截止时间

◆ 期末考试当周的周六晚上22点之前

■ 资料的目录格式

■ U2021刘玉实验报告(主目录, 打包压缩)

◆ 实验1

□ U2021刘玉-实验1实验报告.docx

□ 任务1源代码.zip

□ 任务3源代码.zip

□ 任务7源代码.zip

} 有些任务是选做, 用实际完成任务的编号

◆ 实验2

□ U2021刘玉-实验2实验报告.docx

□ 任务2源代码.zip

□ 任务3源代码.zip

□ 任务6源代码.zip

◆ 实验3

□ U2021刘玉-实验3实验报告.docx

□ 任务1源代码.zip

□ 任务2源代码.zip





1.1 操作系统的定义和特性

操作系统初步认识——基本功能

● 没有安装操作系统的计算机能干什么？



American Megatrends
www.ami.com

AMIBIOS (C) 2006 American Megatrends, Inc.
ASUS PSB-Deluxe ACPI BIOS Revision 0507
CPU : Intel(R) Core(TM)2 CPU 6400 @ 2.13GHz
Speed : 4.01 GHz Count : 2

Press DEL to run Setup
Press F8 for BBS POPUP
Press ALT+F2 to boot from System Recovery
PC2-4300 Dual Channel Interleaved
Initializing USB Controllers .. Done.
2048MB OK

无法继续启动

——普通用户能使用吗？

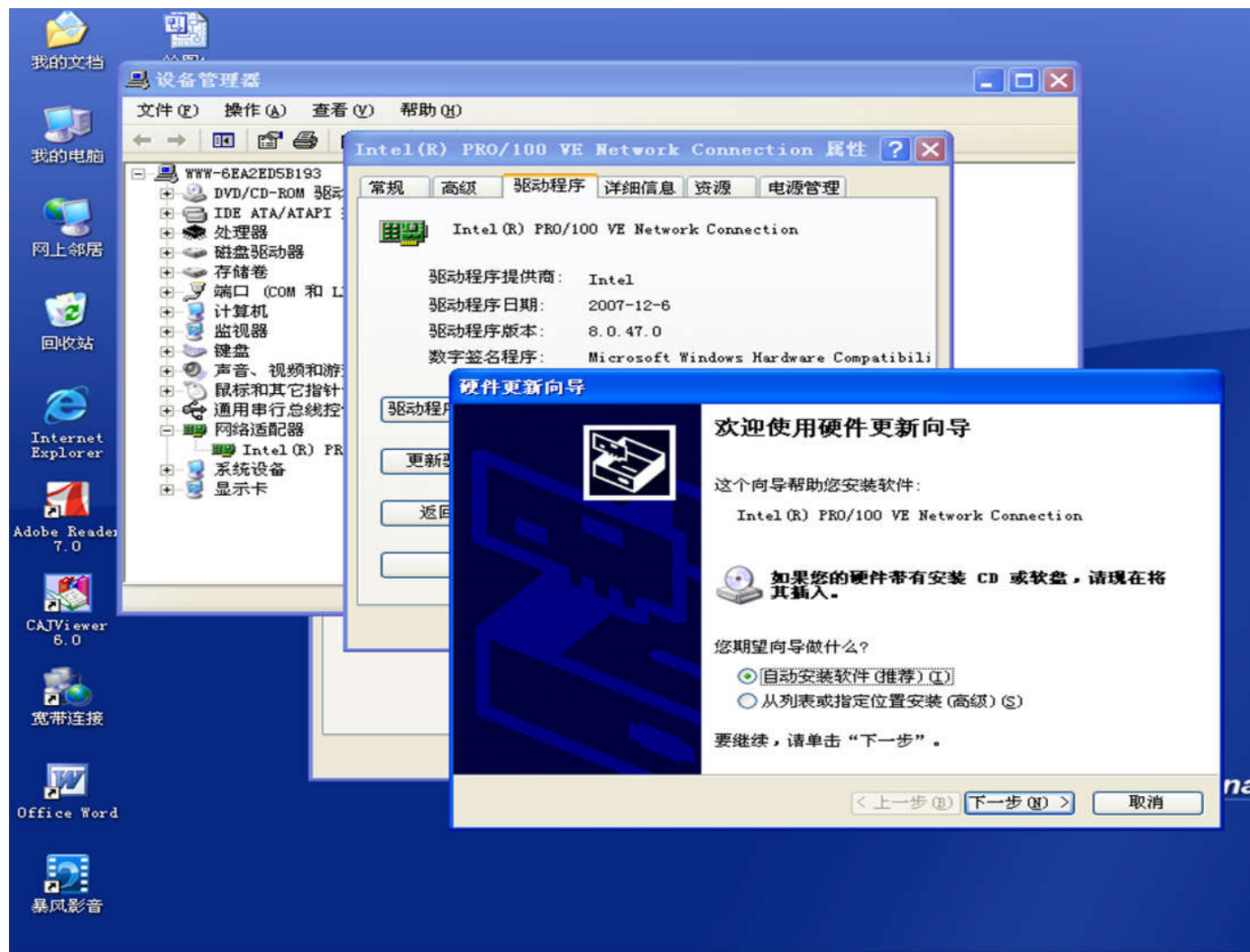
——能运行程序吗？



操作系统初步认识——基本功能

● 基本功能.....

- 提供操作界面
- 控制程序运行
- 管理系统资源
- 配置系统参数
- 监控系统状态
- 工具软件集合



操作系统的认识

● 思考：应用程序运行需要操作系统提供哪些支持？

```
1  //Hello.c → Hello.exe
2  main()
3  {
4      char *Hello = "Hello";
5      printf("%s", Hello);
6      while(TRUE)
7      { //死循环
8          int i = 100;
9      }
10 }
```

- (1) 如何编译/链接？
- (2) **exe**文件放哪里？
- (3) **exe**程序如何启动？
- (4) 如何分配内存？
- (5) **printf**如何输出字符？
- (6) 程序独占**CPU**怎么办？
- (7) 程序结束如何退出系统？

操作系统的定义

- 操作系统是一个大型系统程序

- 操作系统是一个大型的程序系统，它负责计算机系统软件/硬件资源的分配；控制和协调并发活动；提供用户接口，使用户获得良好的工作环境。

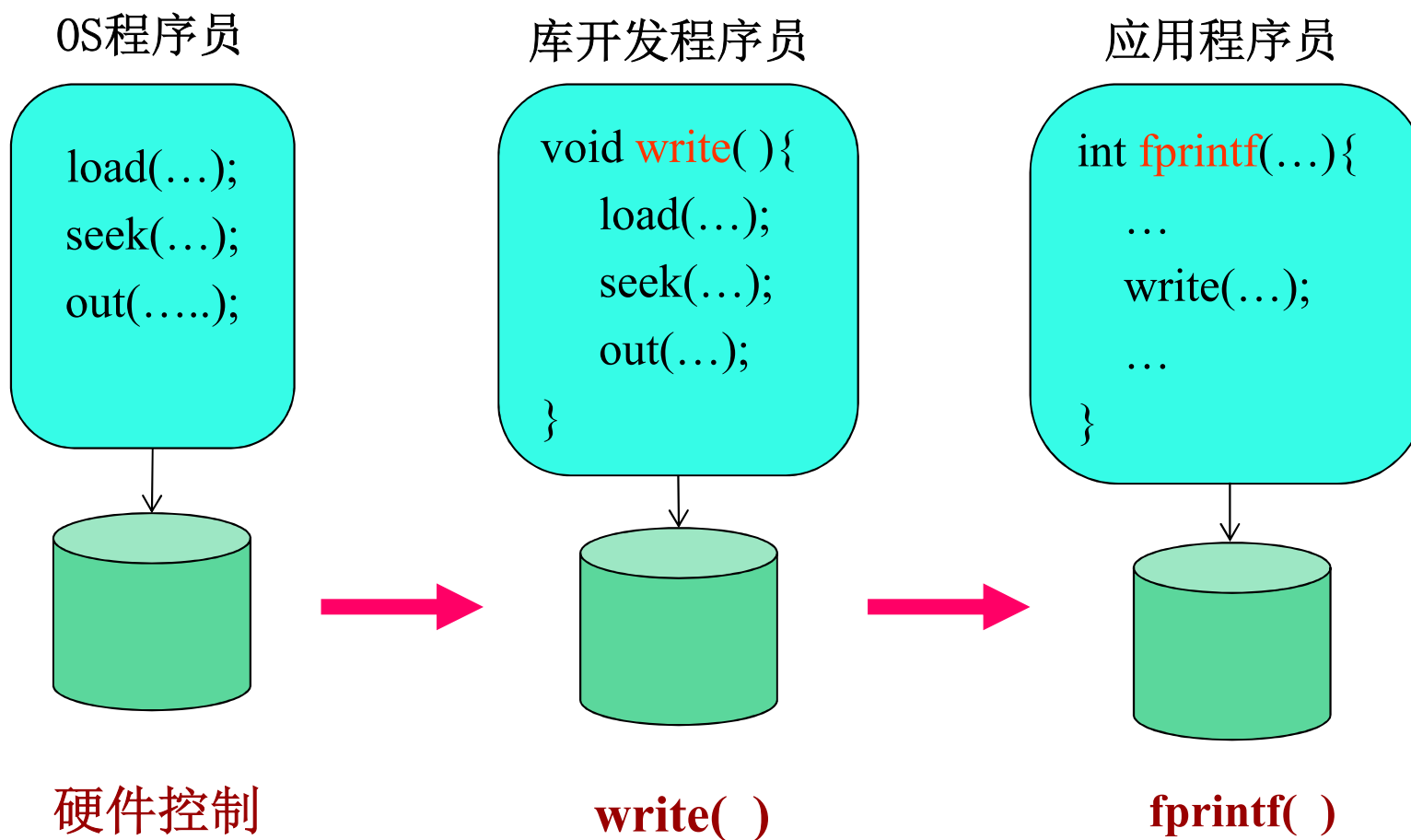
- 两个关键

- 管理并调度资源；

- 为用户提供接口。

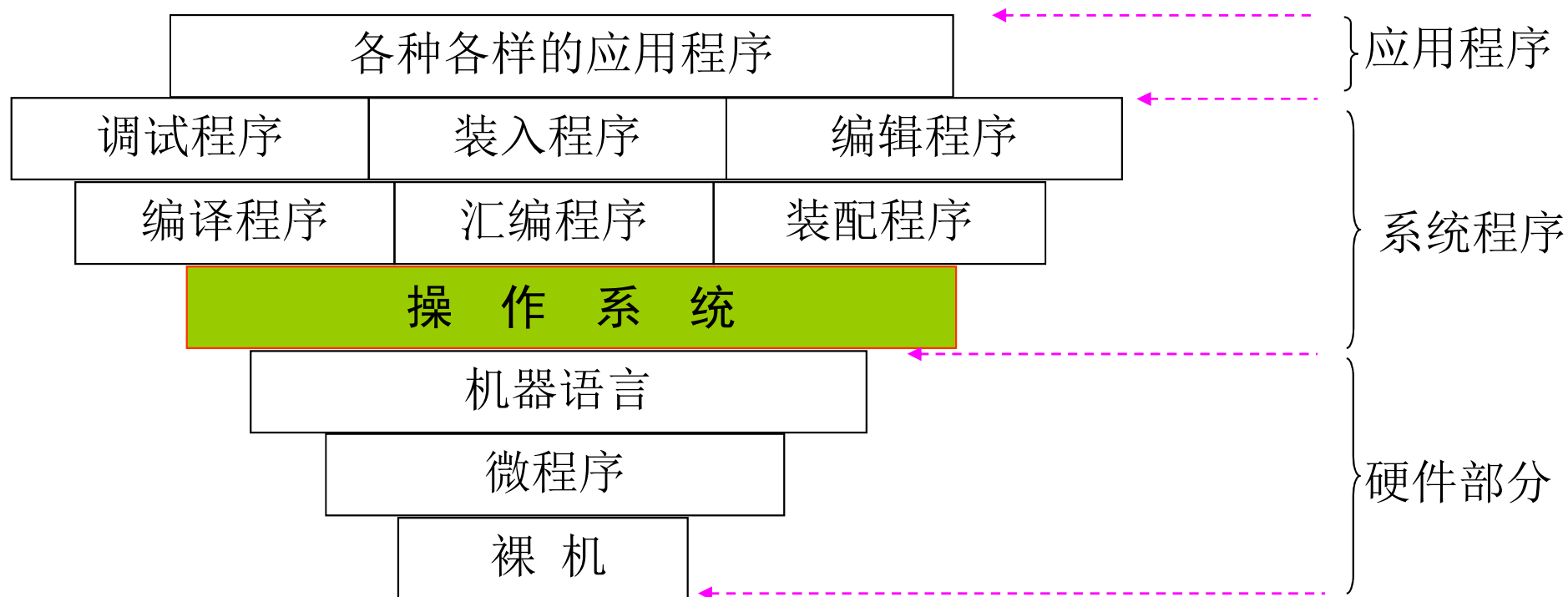
操作系统的定义

- 例子：把数据从内存拷贝到文件 **fprintf()**
- **OS**管理/抽象资源：接口



操作系统的定义

● 操作系统的地位



● 操作系统的特性

■ 并发性

◆ 同时处理多个任务的能力

■ 共享性

◆ 为多个并发任务提供资源共享

■ 不确定性

◆ 具有处理随机事件的能力

□ 中断处理的能力…

□ 自动化能力…

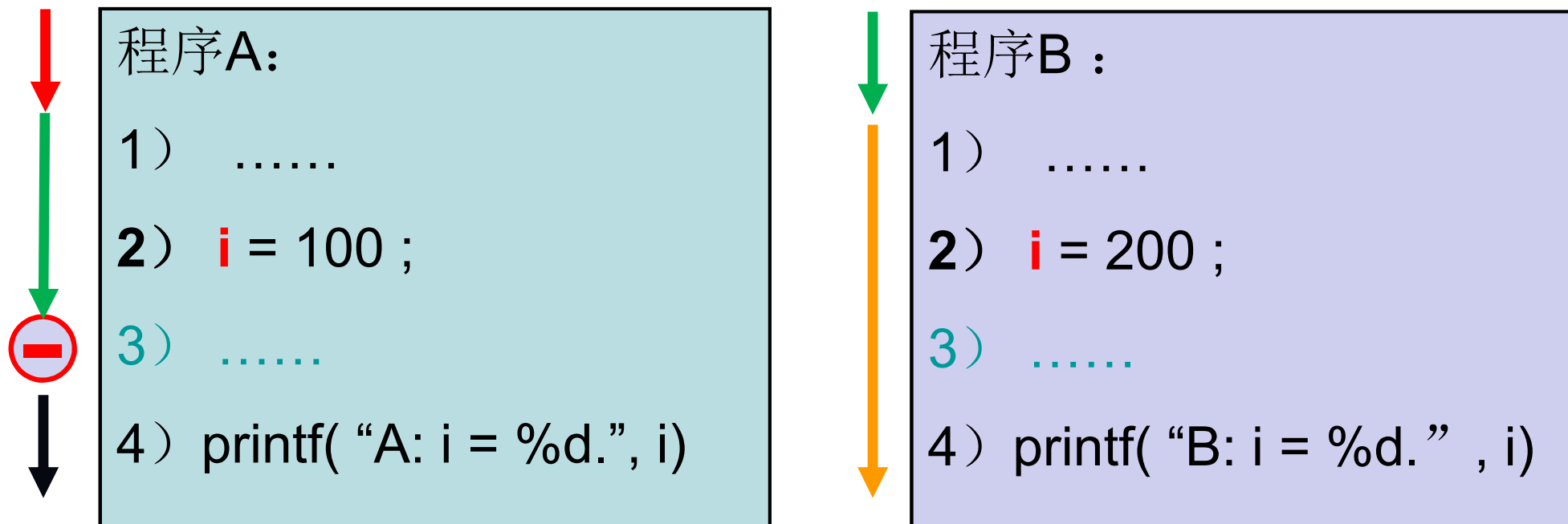


1.2 操作系统的功能

操作系统的功能——并发/分时环境

特点：**OS**会在任何时候和任何地点暂停或继续任何程序的运行。

● **i**是全局变量。请问**A**，**B**并发运行后：**i**输出是多少？



结果 : A: i = 100. ✓

B: i = 200. ✓

结果2: A: i = 200. ✗

B: i = 200. ✓

结果3: A: i = 100. ✓

B: i = 100. ✗

结论: 在并发环境下, “**程序**”不足以描述程序的**运行过程**并确保**运行结果**的正确! → “**进程**”概念

操作系统的功能

● Windows环境下多个“进程”在运行



操作系统的功能

● 操作系统的功能一

■ 进程管理

◆ 处理机分配

◆ 处理机管理

◆ CPU管理

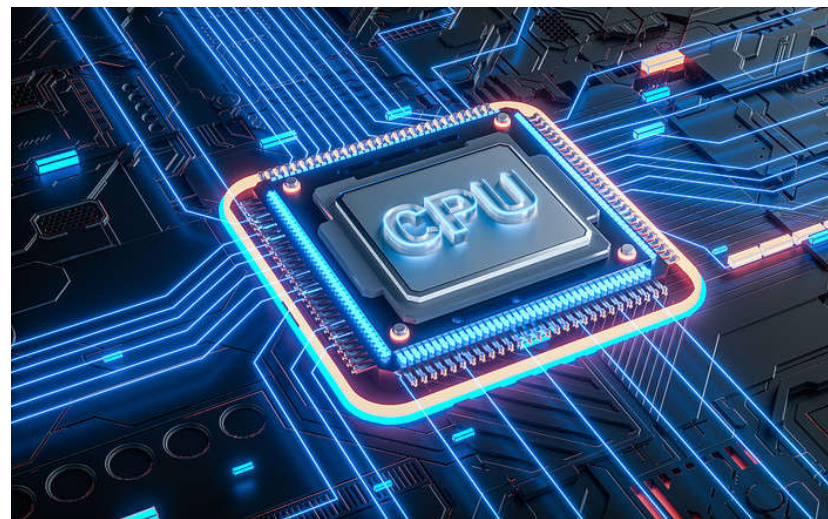
◆ 多个程序如何共享CPU?

■ 具体功能

◆ 进程控制：创建，暂停，唤醒，撤销

◆ 进程调度：调度策略，优先级

◆ 进程通信：进程间通信



案例2：试图耗尽内存的程序

实验与思考：一个试图耗尽内存的程序

while: 循环分配**100M**内存，直到**失败**退出为止！

```
1 int main()
2 {
3     int No = 0;    //序号：循环的序号
4     int * Mem = NULL; //申请的内存指针
5     while (true)
6     {
7         No = No + 1;    //序号递增1
8         Mem = (int *)malloc(1024 * 1024 * 100); //申请100M内存
9         if (Mem == NULL) return 0;                //若申请失败：退出
10        printf("  %3d: 成功申请 100M 内存! \n", No);
11    }
12 }
```

思考: XP/1G内存能循环几次呢？

操作系统的功能

● 操作系统的功能二

■ 存储管理/内存管理

◆ 为应用程序运行分配和管理所需的内存

◆ 支持多道程序设计

■ 具体功能

◆ 内存分配

◆ 内存共享

◆ 内存保护

◆ 内存扩充

◆ 虚拟内存

操作系统的功能

● 操作系统的功能三

■ 设备管理

- ◆ 设备的分配和回收
- ◆ 设备的驱动机制/传输控制
- ◆ 为应用提供统一接口访问设备
 - 设备无关性
- ◆ 高效存取/设备缓冲机制



● 操作系统的功能四

- 文件管理：为用户提供统一的文件存取接口，高效组织存储空间，提高存取效率，实现信息共享和存取控制。

- ◆ 文件用户接口

- ◆ 存储空间管理

- ◆ 文件的操作

- ◆ 目录的操作

- ◆ 存取权限管理

- 文件是设备的抽象



1.3 操作系统的性能

操作系统的性能/评价指标

- 吞吐率

- 在单位时间内处理信息的能力。

- 资源利用率

- 设备(CPU)使用的频度

- 响应能力

- 从接收数据到输出结果的时间间隔。

- 可移植性

- 改变硬件环境仍能正常工作的能力 : 代码修改量

- 可靠性

- 发现、诊断和恢复系统故障的能力。





1.4 操作系统的发展历史

早期的计算机：第一台数字电子计算机ENIAC

- **Electronic Numerical Integrator and Computer**

- 1943年开始建造，1946年投入使用

- 弹道计算，原子弹计算

- **5000次/每秒**

- **18000个电子真空管，**

- **占地182平方米**

- **重量130吨**

- **功耗140kW**



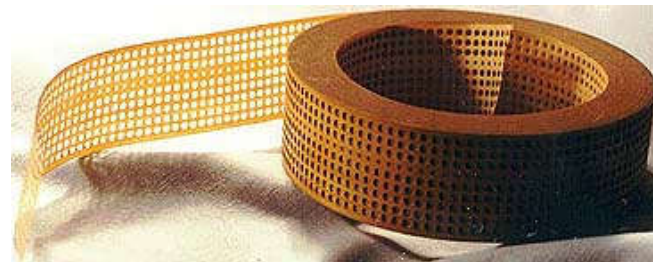
● 操作系统发展的四个典型阶段

- (1) 手工操作(无操作系统) (40年代-50年代初)
- (2) 单道批处理系统 (50年代)
- (3) 多道批处理系统 (60年代初)
- (4) 分时操作系统 (60年代中-至今)

1.手工操作（没有操作系统）

- 电子管时代【1946—1955】

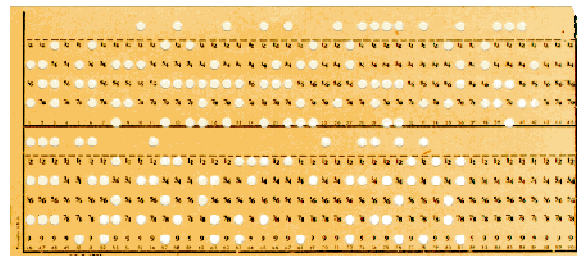
- IBM 701型计算机（1952年，IBM）



- 结构特点

- 硬件：电子管、接线面板（按钮/开关）

- 程序：二进制（卡片/纸带，打孔）



- 使用特点

- 上机：编程（打孔），预约，操作机器（开关/接线）

- 程序启动与结束：手工处理

- 缺点

- 效率低：CPU利用率低 例：100min=50min+10min+40min

装 运行 拆
↓ ↓ ↓

- 用户独占

- 缺少交互

2.单道批处理系统

- 背景

- 晶体管时代【1955—1965】

- 1955年，IBM 推出第一台晶体管计算机：典型机型7094

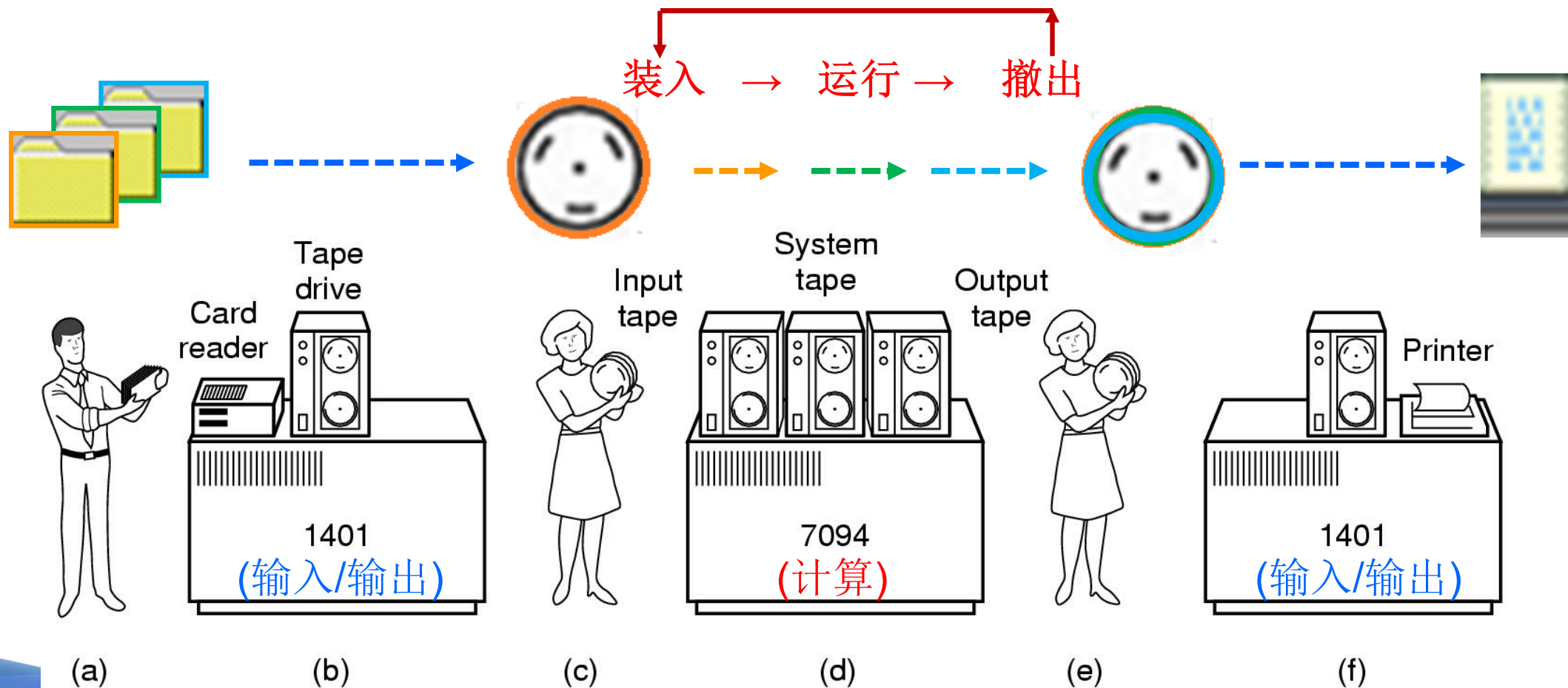


2.单道批处理系统

● 背景

■ 晶体管时代【1955—1965】

■ 1955年，IBM 推出第一台晶体管计算机：典型机型7094



单道批处理系统

● 工作过程

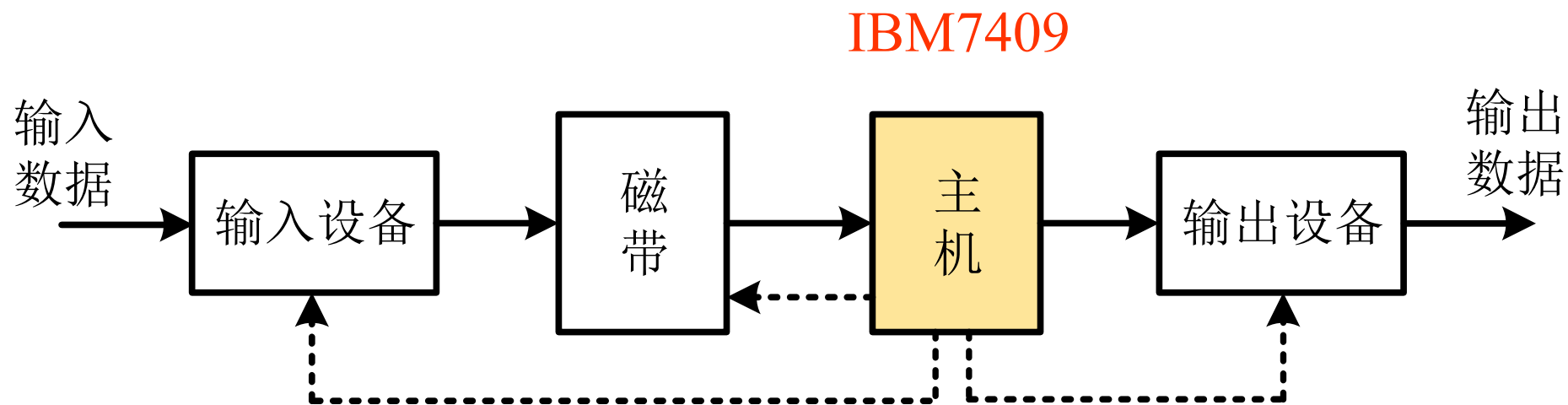
- 管理员将多个作业输入到磁盘形成作业队列;
- 监控程序（操作系统）依次自动处理磁盘中每个作业
 - ◆ 装入—运行—撤出—装入—运行—撤出
- 运行完毕，通知用户取结果

● 工作特点

- 一批：作业队列
- 自动：识别作业
- 单道：串行

单道批处理的两种实现方式

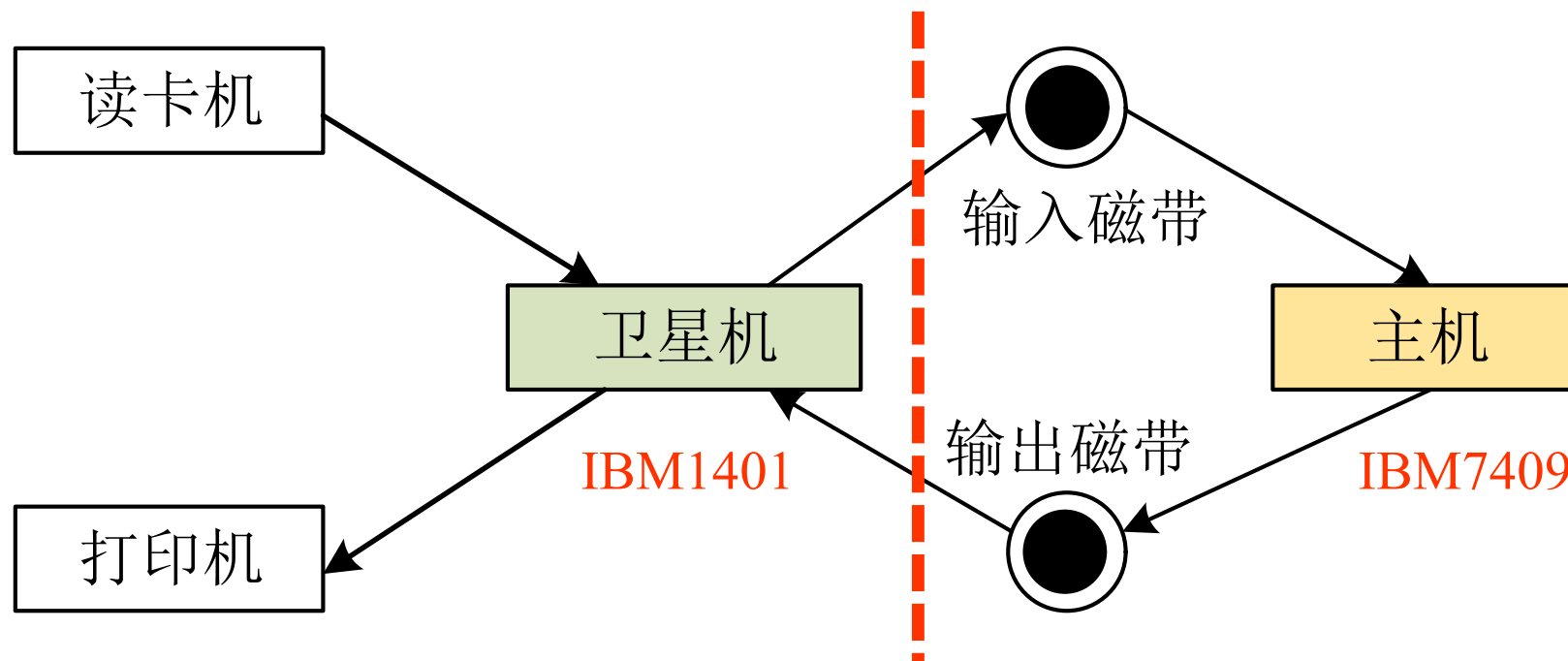
● 方式1：联机批处理



- 特点：主机负责运算，同时控制输入/输出
- 缺点：系统效率低

单道批处理的两种实现方式

● 方式2：脱机批处理

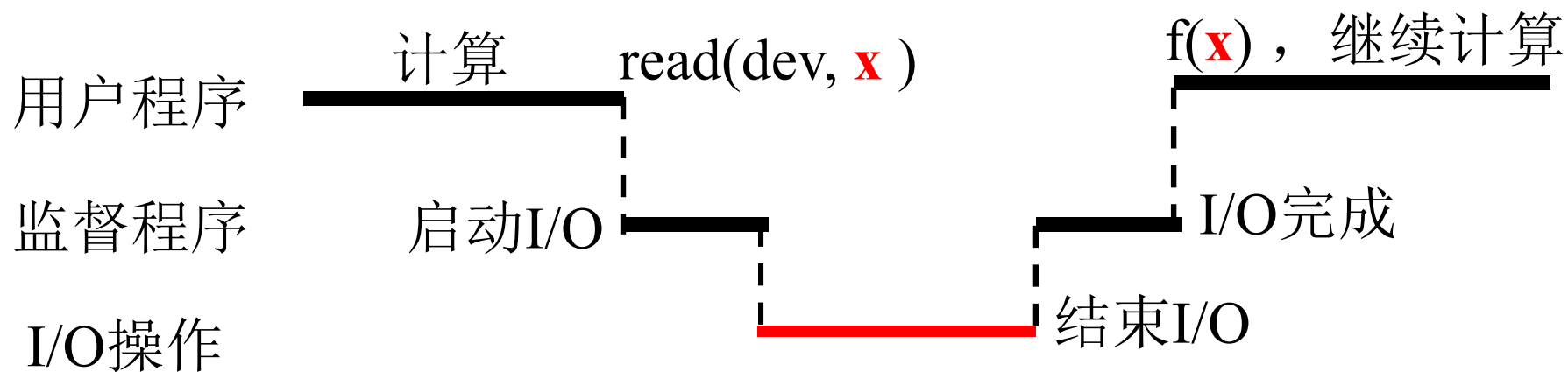


- 特点：主机负责运算；卫星机负责输入/输出。
- 优点：系统效率高
- 缺点：调度不灵活；保护问题

单道批处理系统中CPU利用情况

- 例如：

```
read( dev, x );  
y = f( x );  
.....
```



现象： 外设工作时CPU空闲， **CPU**工作时外设空闲。

结论： **CPU和外设效率低。**

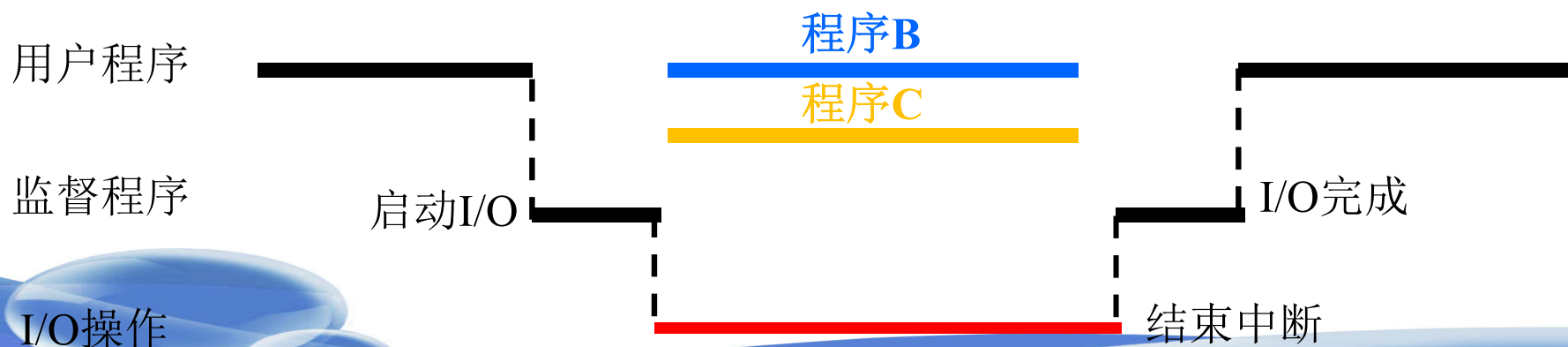
3.多道批处理系统

●多道程序设计技术

- 内存中存放**多道**程序,当某道程序因为**某种原因**（例如请求I/O时）不能继续运行时，**监控程序/OS**便调度另一道程序投入运行，这样使**CPU**尽量处于**忙碌**状态，提高系统效率。

●多道批处理系统

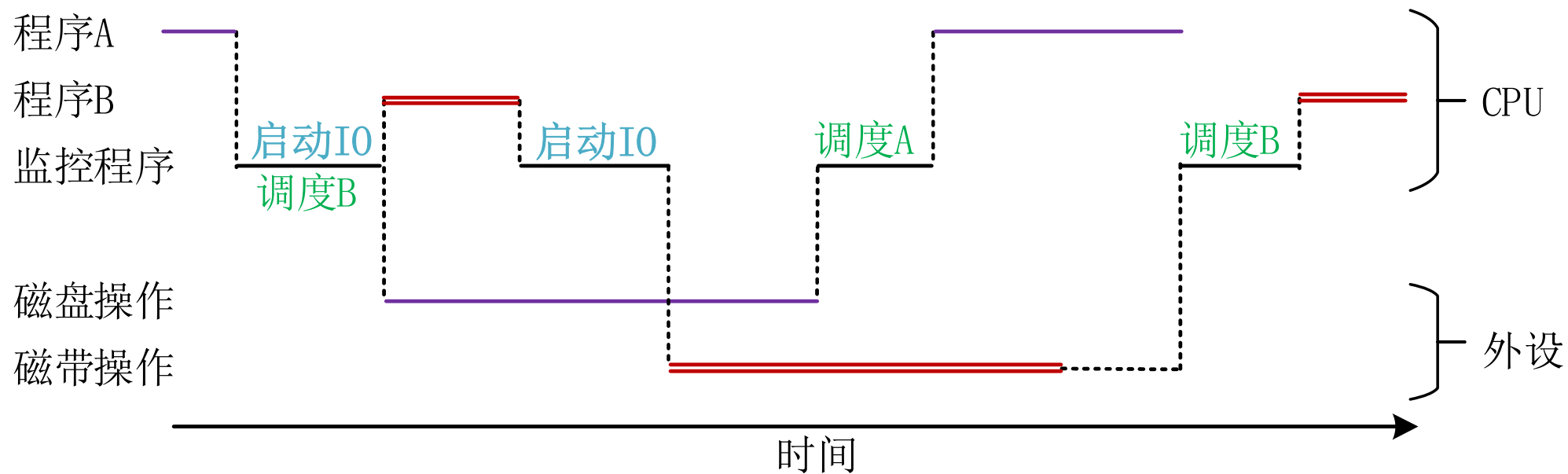
- 采用**多道程序设计技术**实现的处理系统称为**多道批处理系统**。



3.多道批处理系统

● 两个程序的例子

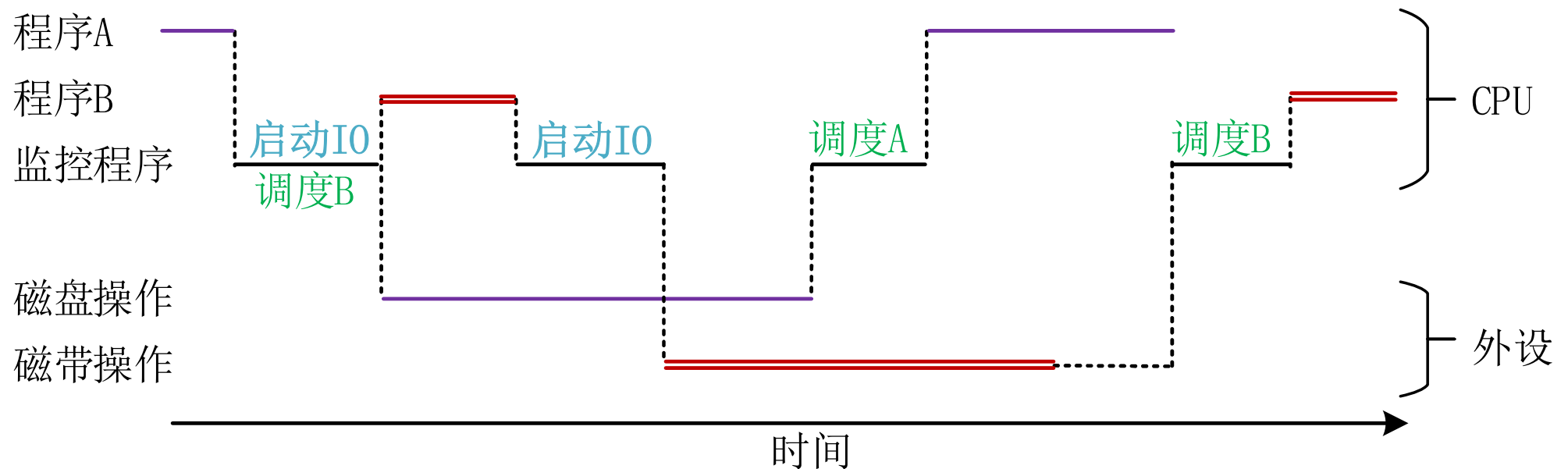
■ 多道程序相互穿插的运行过程



程序A，B相互穿插运行，使CPU和外设都尽量忙碌！

3.多道批处理系统

- 多道批处理系统的设计目的
 - 提高资源的利用率（或吞吐量）
 - ◆ CPU与外设并行
 - ◆ 外设之间也并行



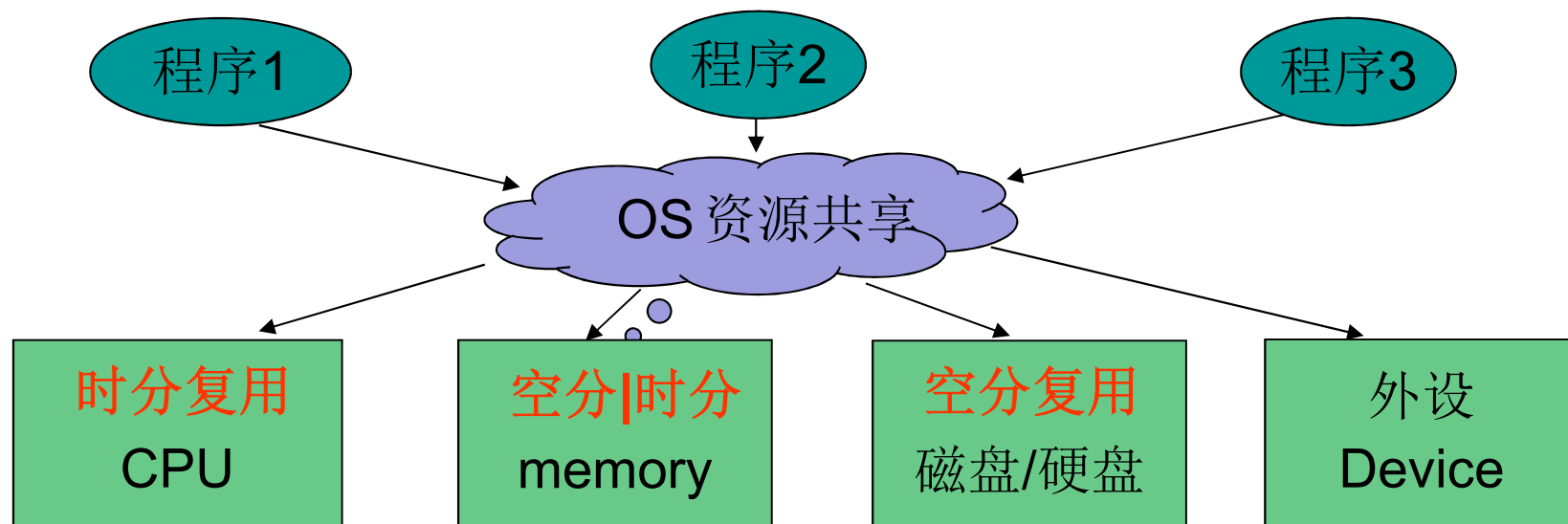
多道程序设计思想

- 实现资源共享

- 时分：分成多个时段：不同进程错开时段使用。

- 空分：分成多个单元：不同进程使用不同单元。

- 多个程序同时在计算机/虚拟机上运行



多道批处理系统的特点

- 多道

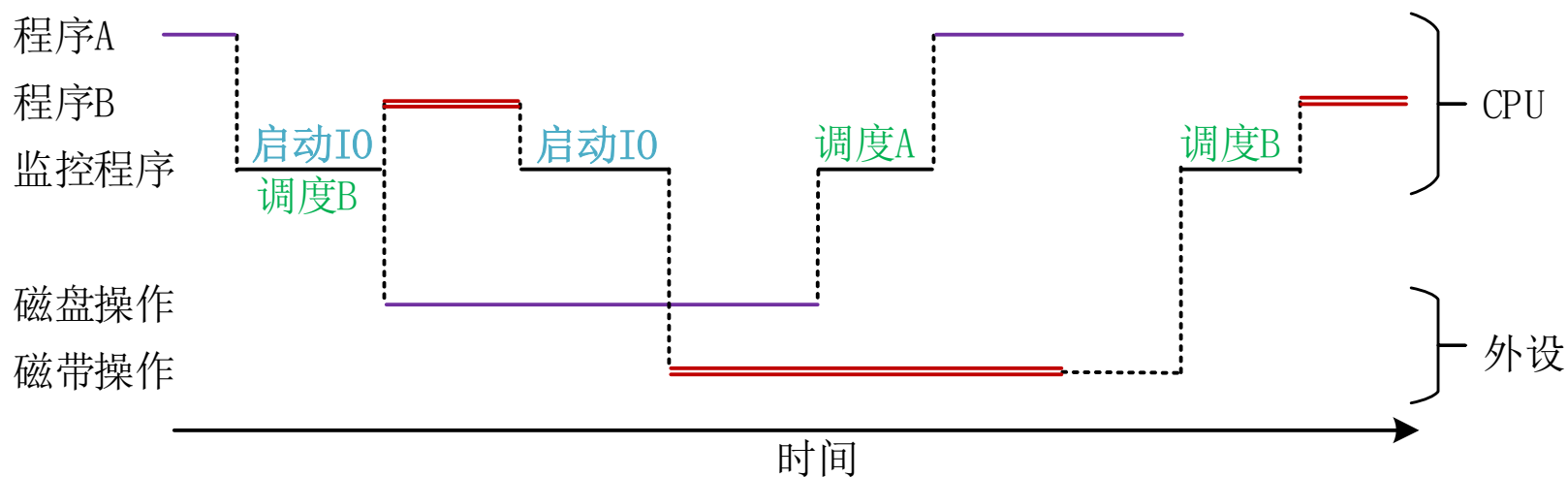
 - 内存同时存放多道程序

- 并行

 - 宏观上

- 串行

 - 微观上



- 意义：多道技术是现代操作系统的雏形。

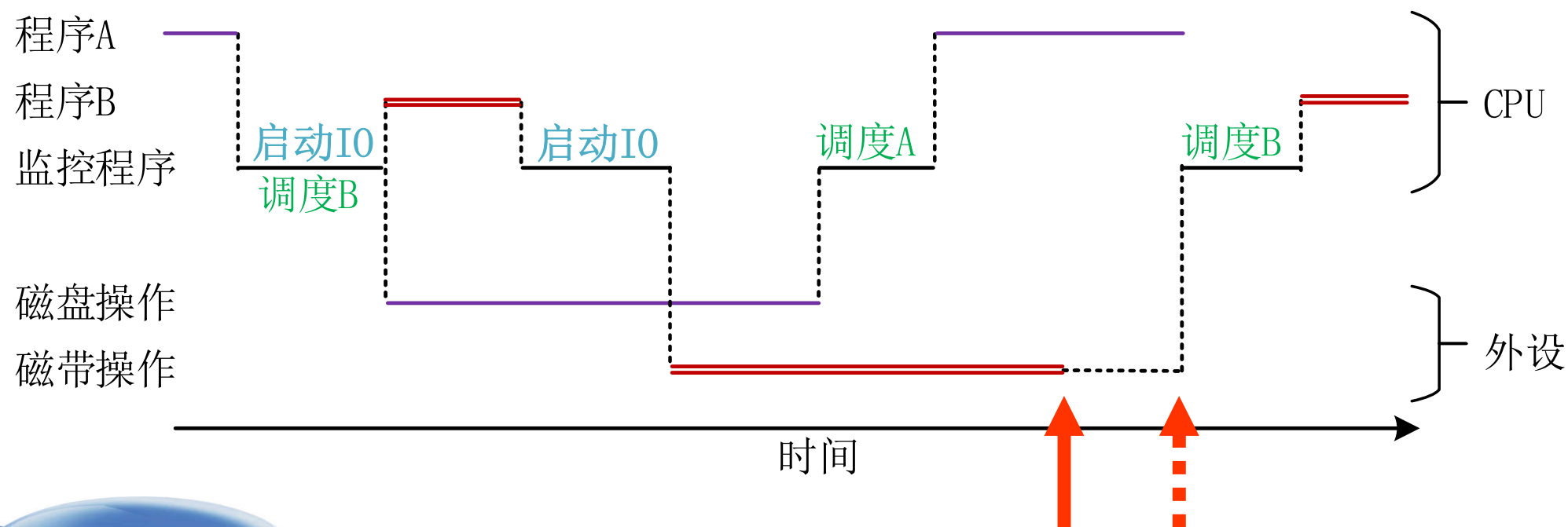
多道批处理系统的缺点

● 缺点

■ 作业处理时间长

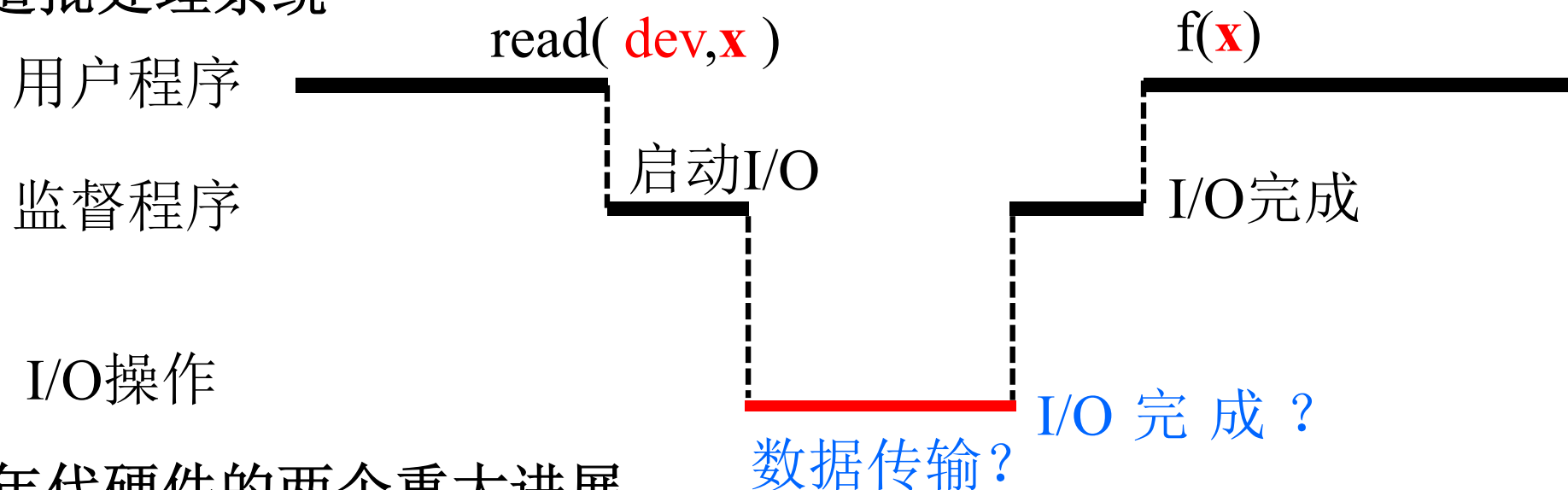
■ 运行过程不确定

■ 交互能力差



60年代硬件的两个重大进展

- 多道批处理系统



- 60年代硬件的两个重大进展

- 中断技术

- ◆ CPU收到外部信号（中断信号）后，转去处理外部事件，处理完毕后回到中断处（断点）继续原来工作。

- 通道技术

- ◆ 专门处理外设与内存之间数据传输的处理机。

4.分时操作系统

● 背景

■ 事务性任务和程序的涌现

◆ 交互性高

◆ 响应快速

◆ 多任务/多用户

■ 硬件

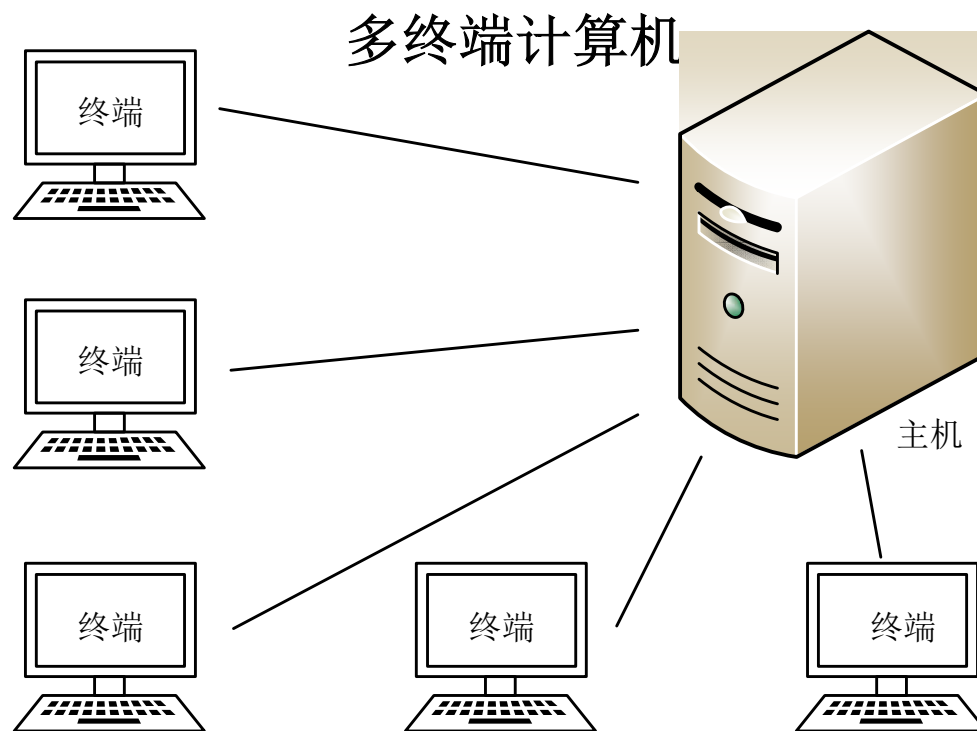
◆ 中断技术

◆ 大规模集成电路

◆ 多终端计算机

□ 用户通过终端连接主机，通过终端与主机交互。

□ 主机采用**分时技术**轮流为每个终端服务。



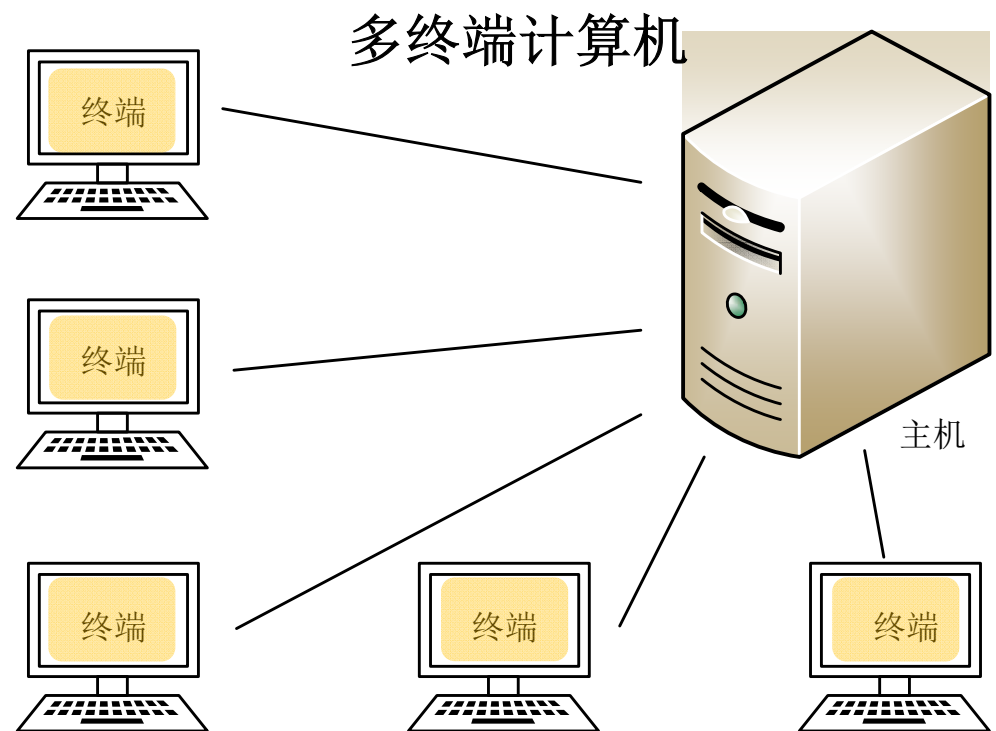
分时技术

- 分时技术概念

- 主机以很短的“时间片”为单位，把CPU轮流地分配给每个作业（终端/用户）使用，直到全部作业被运行完。

- 关键点

- 时间片：较短时间间隔
- 循环/轮流：响应及时
- ◆ 独占主机



● 分时系统的特点

■ 多路调制性

◆多用户联机使用同一台计算机

■ 独占性

◆用户感觉独占计算机

■ 交互性

◆及时响应用户的请求

● 分时操作系统的实例

■ Linux:50ms或可设置

■ Windows:?

■ CTSS (Compatible Time Sharing System)

◆MIT开发 200ms

大型分时系统的实践：Multics 项目

- 项目基本信息

- 1962年，ARPA支持

- 技术参与：MIT、BELL和G.E

- 开发一种“公用计算服务系统”

- ◆ MULTiplexed Information and Computing Service

- ◆ 支持波士顿地区所有分时用户

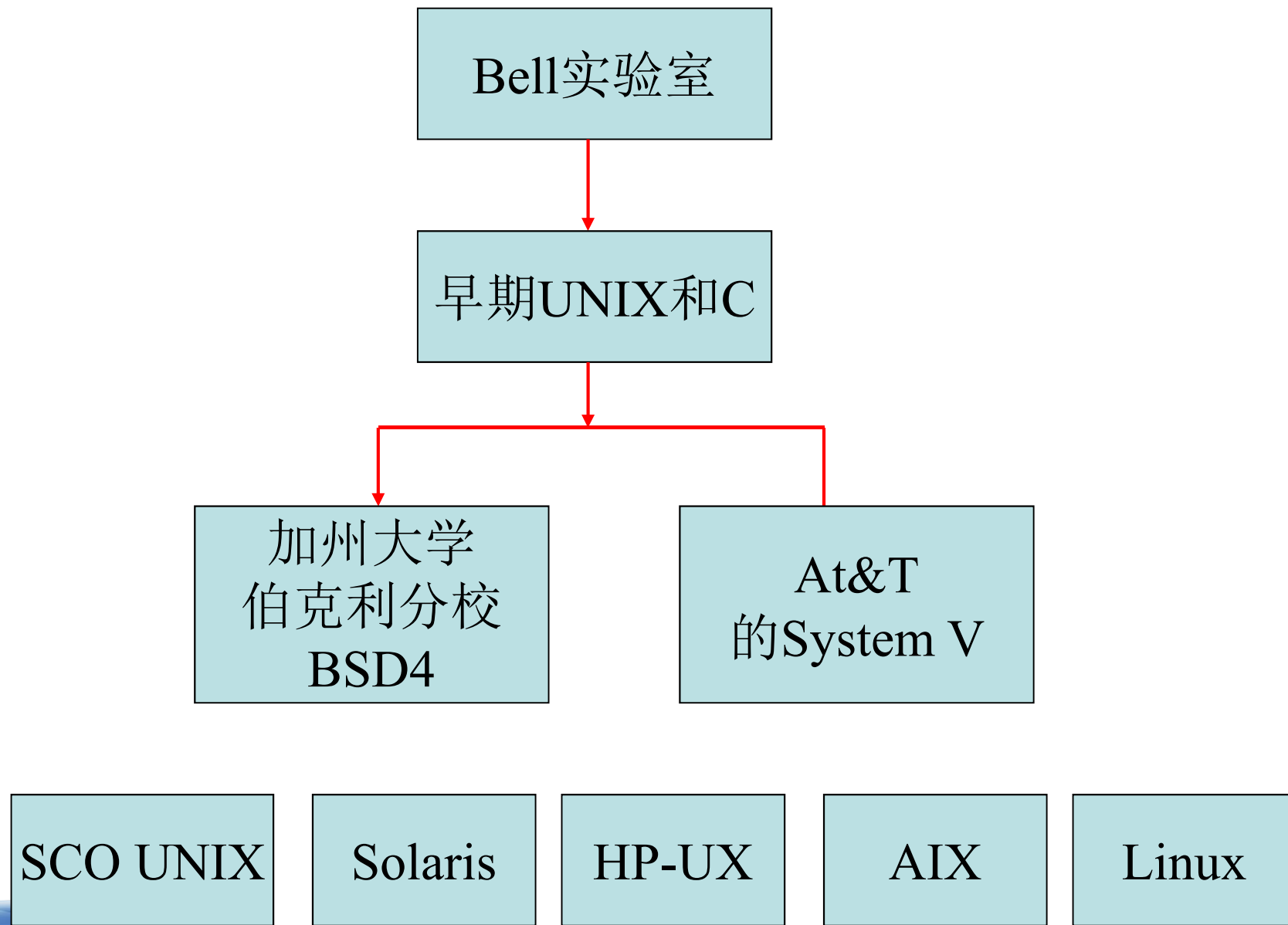
● Multics设计目标

- 使用远程终端/电话线接入计算机主机
- 高可靠的大型文件系统
- 支持数字运算和分时系统的各种应用
- 多种程序设计环境和人机界面
- 连续工作（无关机）
- 可变的配置能力，无需用户程序重新配置
- 大容量的用户信息共享
- 存储和构造层次化信息结构的能力
- 允许随技术的发展，而不断进化系统

UNIX操作系统

- 第一个实用化的分时操作系统
 - 集成各种先进概念和技术的操作系统
- 革新和创造
 - 实现了操作系统的可移植性
 - 与计算机硬件无关性
 - “特殊文件”（Special File）： 外设看作文件。
- 促使UNIX系统成功的因素
 - C语言编写，可移植
 - ◆ 笔记本、PC机、工作站、巨型机
 - 源代码高效，容易适应特殊需求
 - 良好，通用、多用户、多任务、分时的操作系统

UNIX操作系统



● 思考题1

- 多道批处理和分时系统在内存中都有多个作业同时存在，且CPU都会在作业间进行切换。请问这两种切换有什么区别？

● 操作系统的进一步发展（分时系统的衍化）

- 实时操作系统/嵌入式操作系统
- 微机操作系统（PC机操作系统）
- 多处理机操作系统
- 网络操作系统
- 分布式操作系统
- 嵌入式操作系统

- 产生背景

- 实时要求

- ◆ 实时事务：军用电子、工业控制，智能仪器等

- ◆ 某些任务要**优先紧急地**处理

- 强调作业完成的时限(dead-line)

- ▲ 完成时间的可预测性

- 实时系统的分类

- 硬实时系统

- 软实时系统

实时系统的分类

- 硬实时系统：必须限时完成

- 火炮控制系统
- 航空航天
- 制导系统
- 目标识别和跟踪
- 工业控制
- 汽车电子系统
-



近地报警系统是一个实时系统

Ground Proximity Warning System

实时系统的分类

- 软实时系统：尽可能快完成

- 网络视频
- 互动网游
- 广播
- 通讯
-



实时系统的分类

- 典型实时操作系统

- vxWorks
- LiteOS
- RT-Thread
- ucOS-II
- rtLinux
-



- 嵌入式操作系统

- 嵌入式系统使用的操作系统
- 实时操作系统 \in 嵌入式操作系统

个人计算机时代的微机操作系统

● CP/M 操作系统

- 1973年Gary Kildall设计

 - ◆ Control Program/Microprocessor

- 良好的层次结构

 - ◆ BIOS+OS

 - ◆ 可移植性好

- 适应性好，易学易用

- 1980年初流行最广的8位操作系统之一。

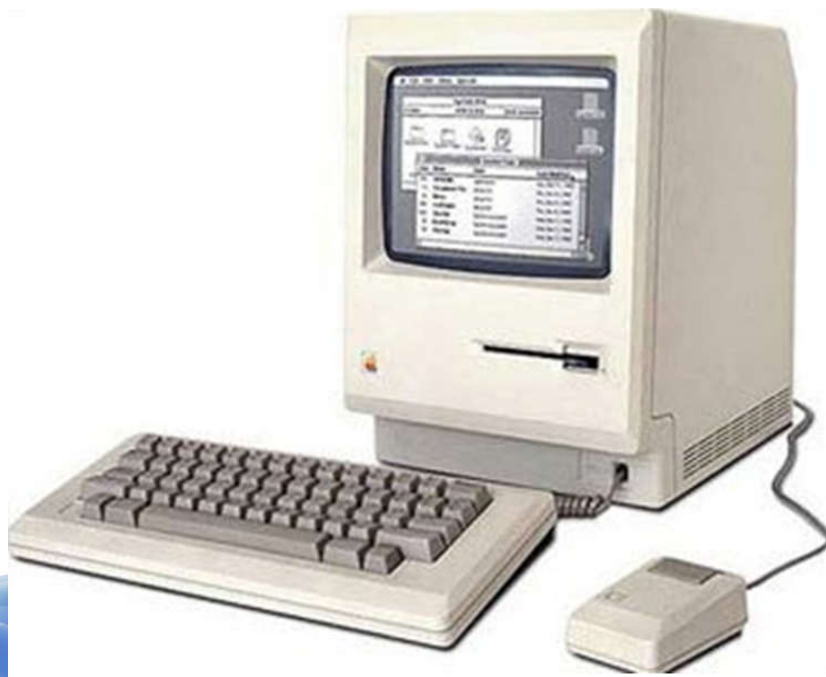


个人计算机时代的微机操作系统

- 苹果(1976年 乔布斯, 沃兹)

- Macintosh和MAC OS

- ◆ 1984年, 图形界面和鼠标
 - ◆ Macintosh系列计算机专用
 - ◆ 首个在商用领域成功的图形用户界面。



个人计算机时代的微机操作系统

- 微软的 MS DOS

- 磁盘操作系统 (Disk OS)

- 单用户单任务

- 1980年IBM开始生产微型计算机IBM PC。

- 微软公司提供MS DOS预装在IBM PC上。



个人计算机时代的微机操作系统

- 微软的Windows操作系统

- 1985年11月，发布Windows 1.0

- 1992年04月，发布Windows 3.1

- 1993年05月，发布Windows NT

- Windows 95/98

- Windows 2000

- Windows XP

- Windows 7

- Windows 10/11

多处理机系统

● 定义

- 包含两个或多个功能相当的处理器（CPU）
 - ◆ 例：SMP(Symmetrical Multi-Processor,对称多处理机)
- 共享：内存，I/O通道、外设
- 单一操作系统控制

● 特点

- 具有一定的并行处理能力
- Linux支持SMP
 - ◆ CONFIG_SMP配置项

网络操作系统

● 网络操作系统

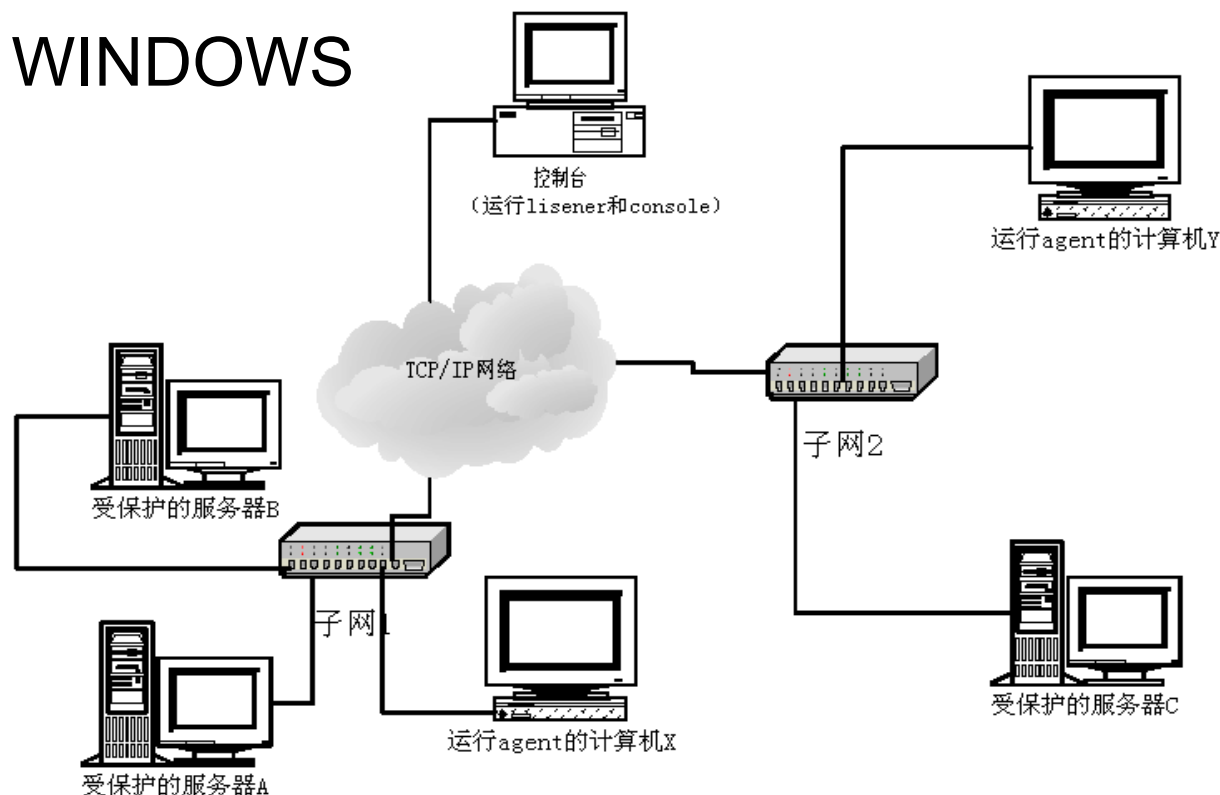
■ 计算机网络：通过协议将多台计算机互连实现资源共享和信息传递的计算机系统。

■ 网络操作系统=普通操作系统+网络通信+网络服务。

◆ 例：UNIX/ LINUX/ WINDOWS

● 特点

- 主机独立自治
- 通信协议
- IP/端口的标识



● 功能/定义

- 分布式系统：通过**网络**将多个**通用资源部件**互联，并能对资源进行**全局统一管理**和**调度**的计算机系统。

● 特点

- **可运行分布式程序**
- 主机自治又相互协调
- 可扩展性 | 高可靠性

● 与计算机网络的区别

- 支持单点登录（虚拟为“单台”计算机）
- 支持资源透明存取（计算机网络：通过**IP**指明主机）
- 支持合作运算（计算机网络：不支持）

国内操作系统的研制历史

- 60年代末至70年代初，杨芙清院士主持
 - 我国第一台百万次集成电路计算机 **(150) 操作系统**
 - 支持多道程序运行
 - 在石油勘探领域成功应用
- 70年代中后期，杨芙清院士主持
 - 我国第一个高级语言书写的**DJS240机**操作系统DJS200/XT2
- **GX73多机实时操作系统**（1978年）国防科技大学
 - 1980年装在“远望-I号”航天测量船上
 - 发射运载火箭、潜水艇水下发射的测控任务；
 - 同步地球卫星的测控、定轨、控制任务
- **YHOS巨型操作系统**（1983年）国防科技大学
 - 用于YH-1、YH-2超级计算机
 - 石油勘探、天气预报和核物理研究

国内操作系统的研制历史

● COSIX 国产UNIX类操作系统

■ Chinese Operating System like UNIX

■ 国家八五、九五重点科技攻关成果

■ 以中软为首，联合国内18个单位共同完成

◆ 初始阶段：1989年～1993年

◆ 发展阶段：1994年～1995年

◆ 完善阶段：1996年～1997年

● Hopen OS（女娲计划）

■ 北京凯思昊鹏软件工程有限公司

■ 1999年3月推出“女娲计划”

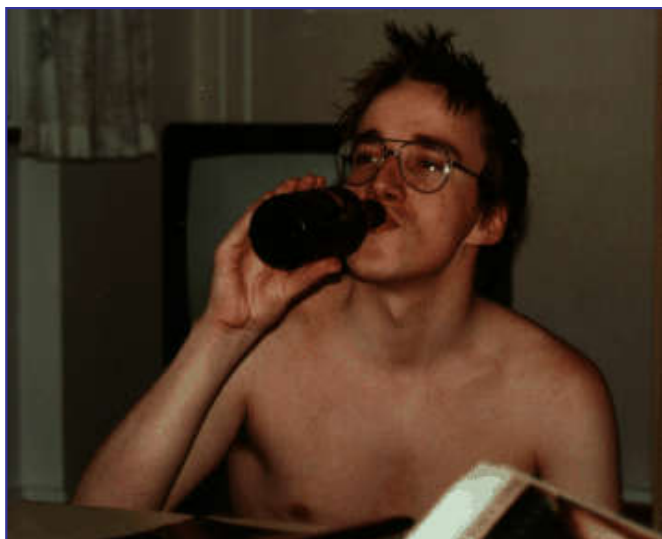
■ 完全国产化的嵌入式操作系统

推荐学习的大型开源操作系统



■ Linus Torvalds[芬兰.赫尔辛基大学],1991.10

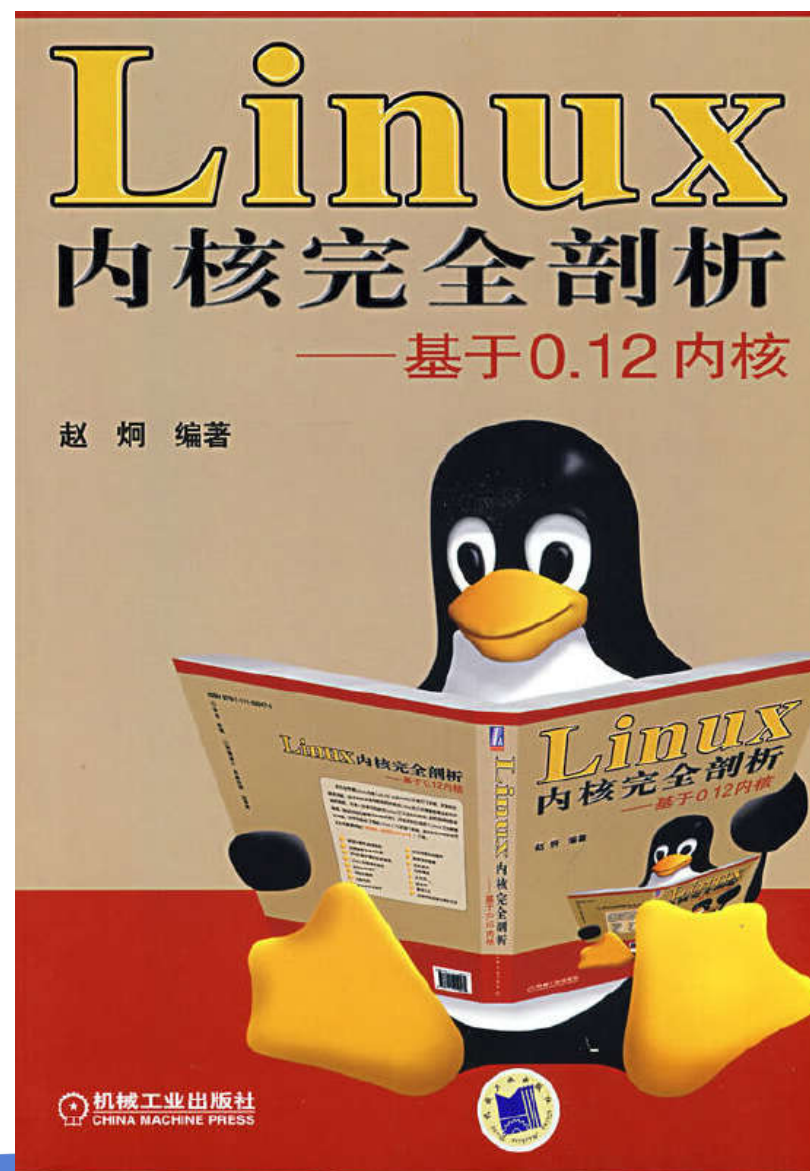
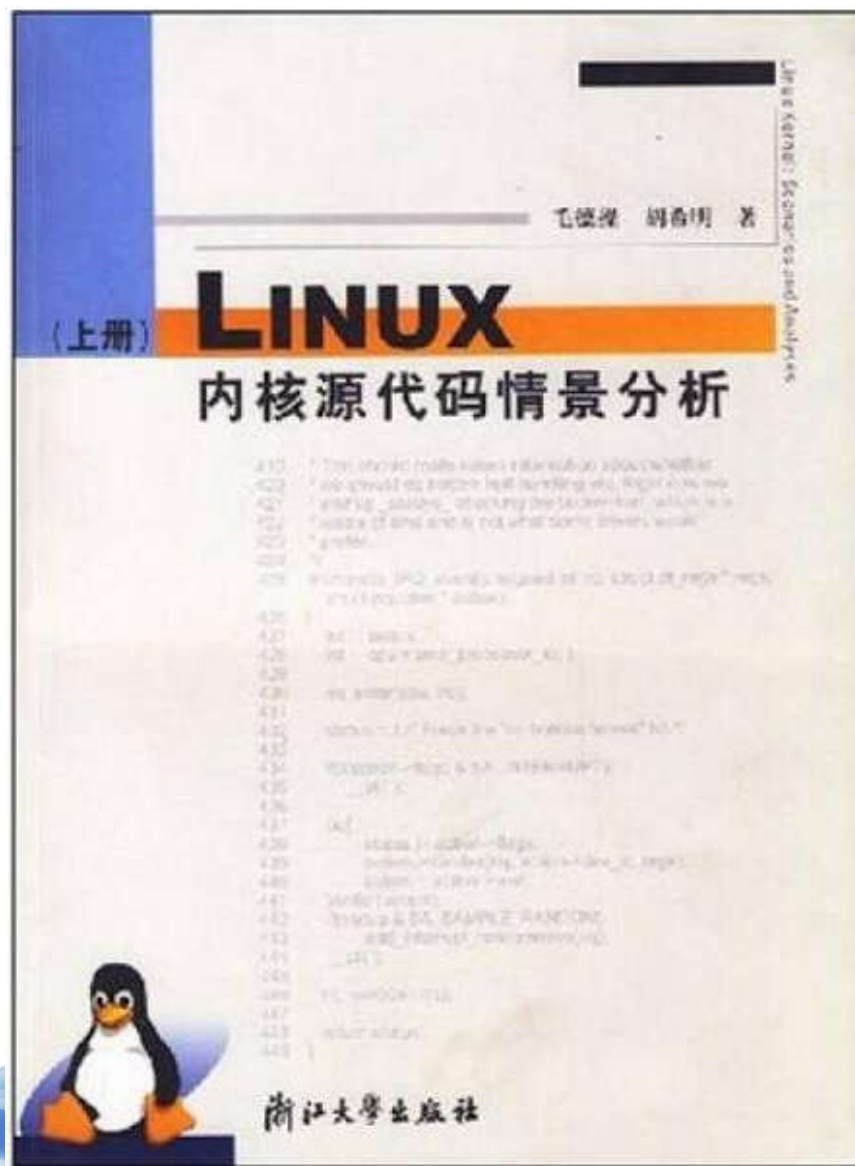
■ 1994年3月, Linux1.0版正式发布



```
Hello everybody out there using minix-  
I'm doing a (free) operation system (just a hobby,  
won't be big and professional like gnu) for 386(486) AT clones.
```


适合学习和研究的简化操作系统

● Linux0.11/Linux0.12



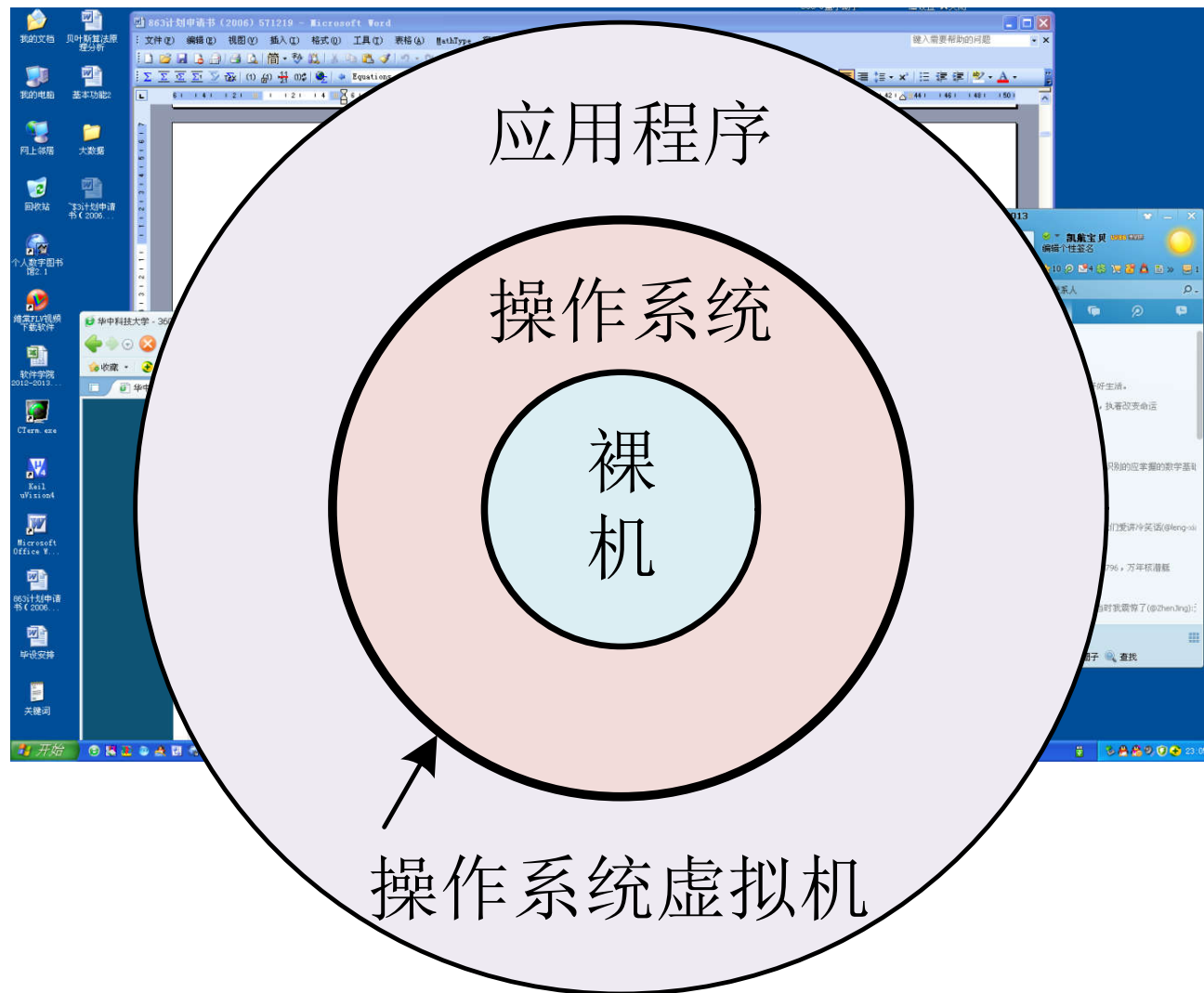


1.7 操作系统虚拟机

操作系统“虚拟计算机”的概念

- 面对用户，裸机配置操作系统后称为**操作系统虚拟机**

- 用户界面
- 屏蔽硬件细节
- 扩展硬件功能
- 系统更安全
- 系统更可靠
- 效率更高





1.8 操作系统的逻辑结构

操作系统的逻辑结构

- 逻辑结构

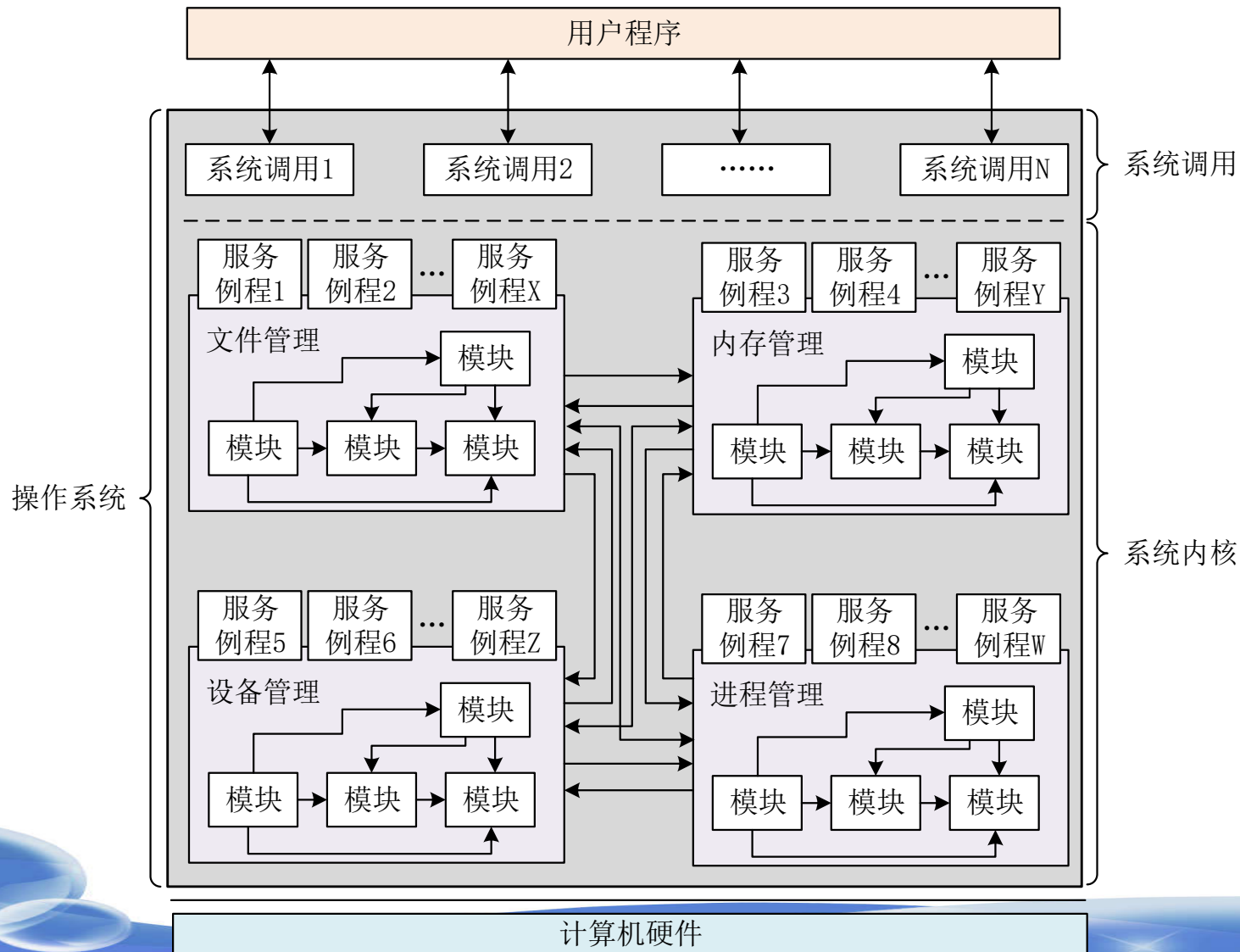
- OS的设计和实现思路

- 逻辑结构的种类

- (1) 整体式结构(单体式结构, 宏内核结构)
 - (2) 层次式结构
 - (3) 微内核结构

(1) 整体式结构

- 以**模块**为基本单位构建
 - 每个模块具有特定的功能



(1) 整体式结构

● 定义

■ 模块化结构/单体内核结构/宏内核结构

■ 操作系统由大量过程构成。每个过程都有明确参数列表、返回值类型。大多数过程是可以相互间调用。

● 优点

■ 模块的设计、编码和调试独立。

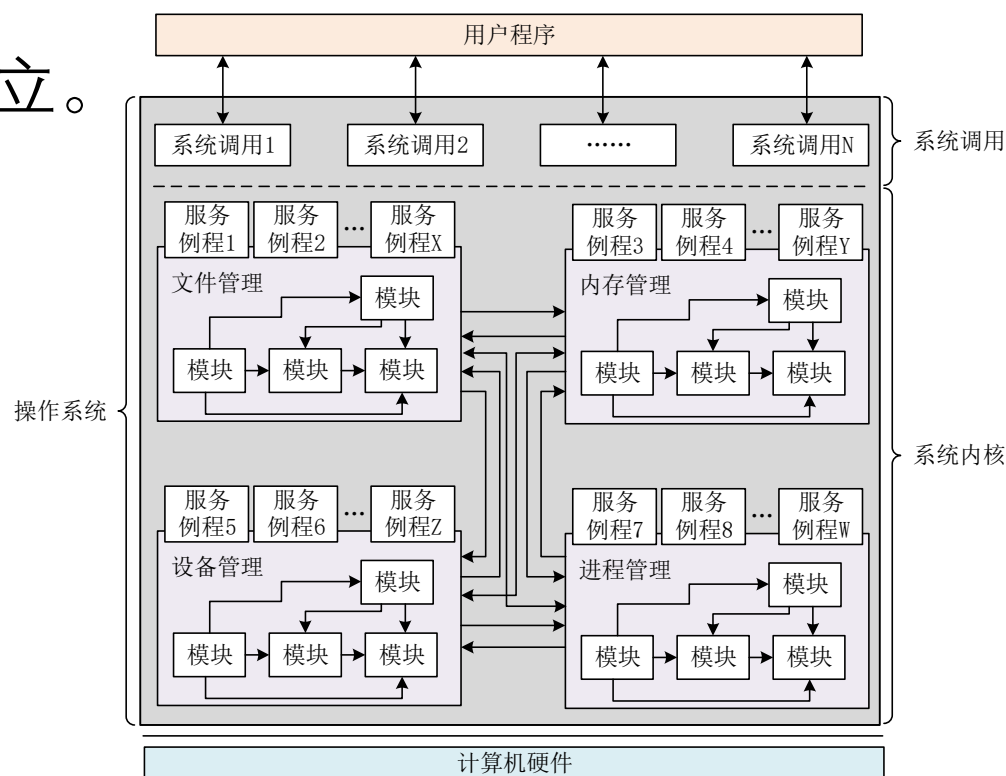
■ 模块之间可以自由调用。

● 缺点

■ 错误容易扩散

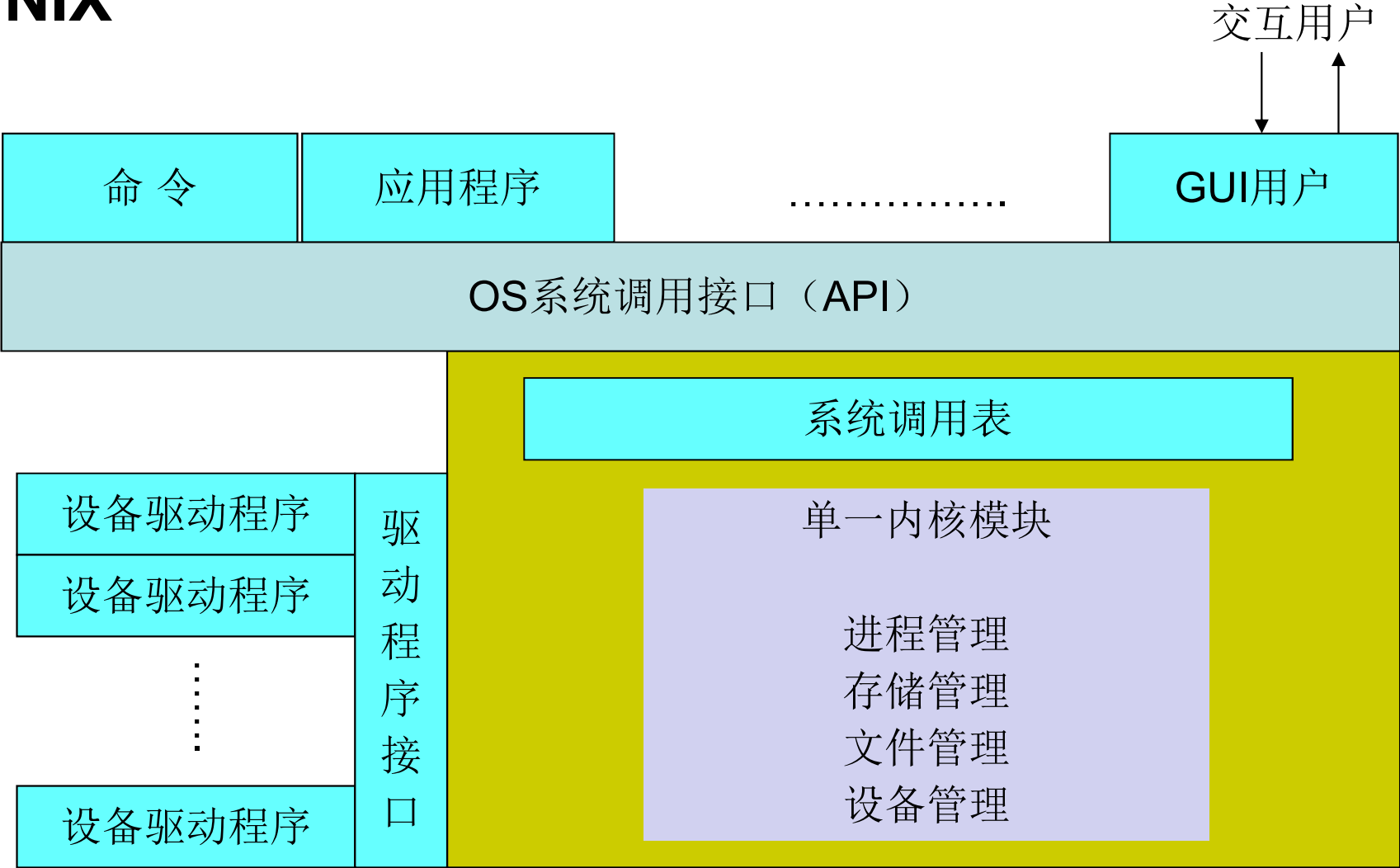
■ 开发和维护困难

■ 可伸缩性差



整体式OS实例

● UNIX



● UNIX

- 宏内核，单体内核
- 对外提供一组系统调用
- 设备驱动与内核其它部分分开

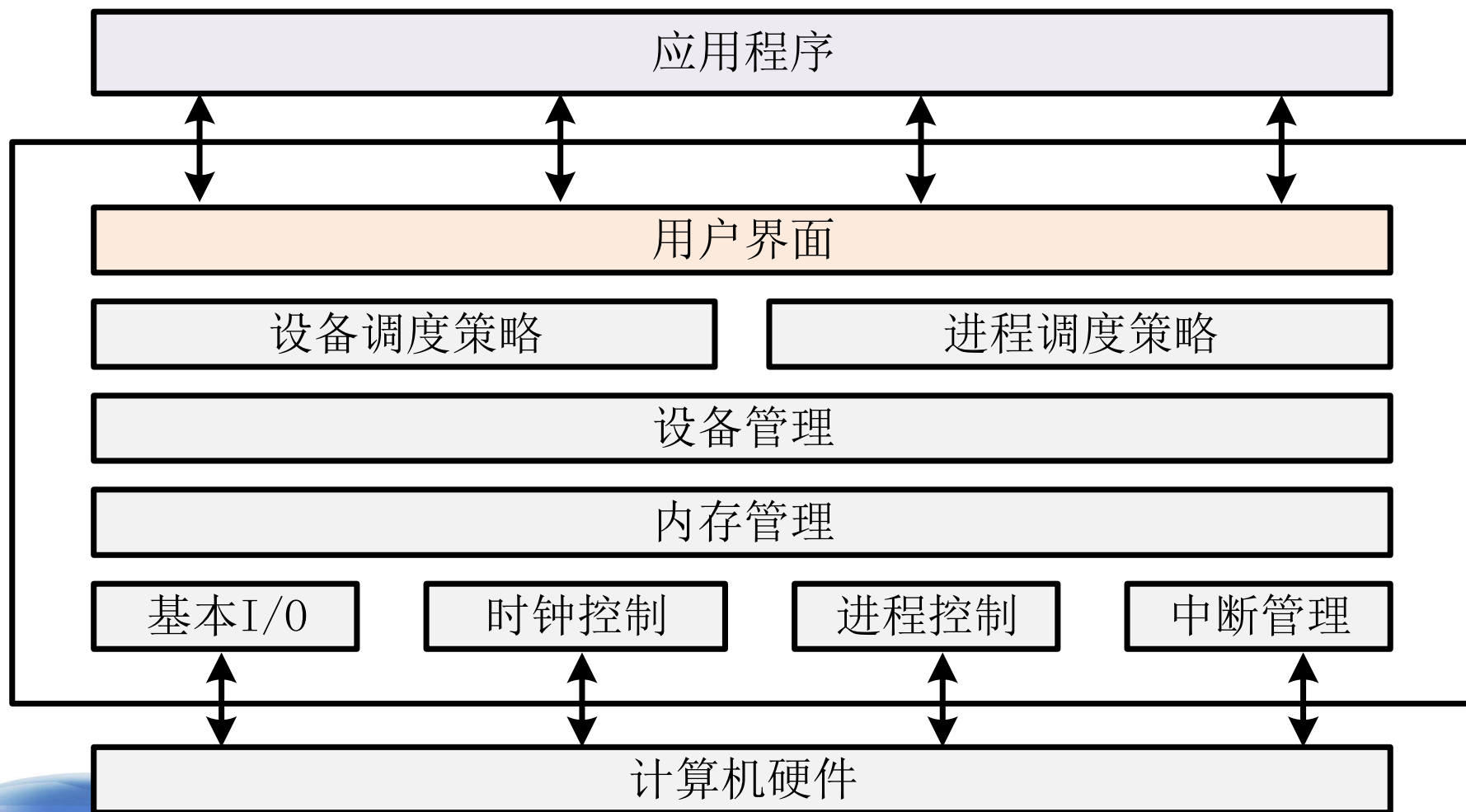
● Linux

- 宏内核，单体内核
- 对外提供一组系统调用
- 支持动态可安装模块

分层结构的OS

● 定义

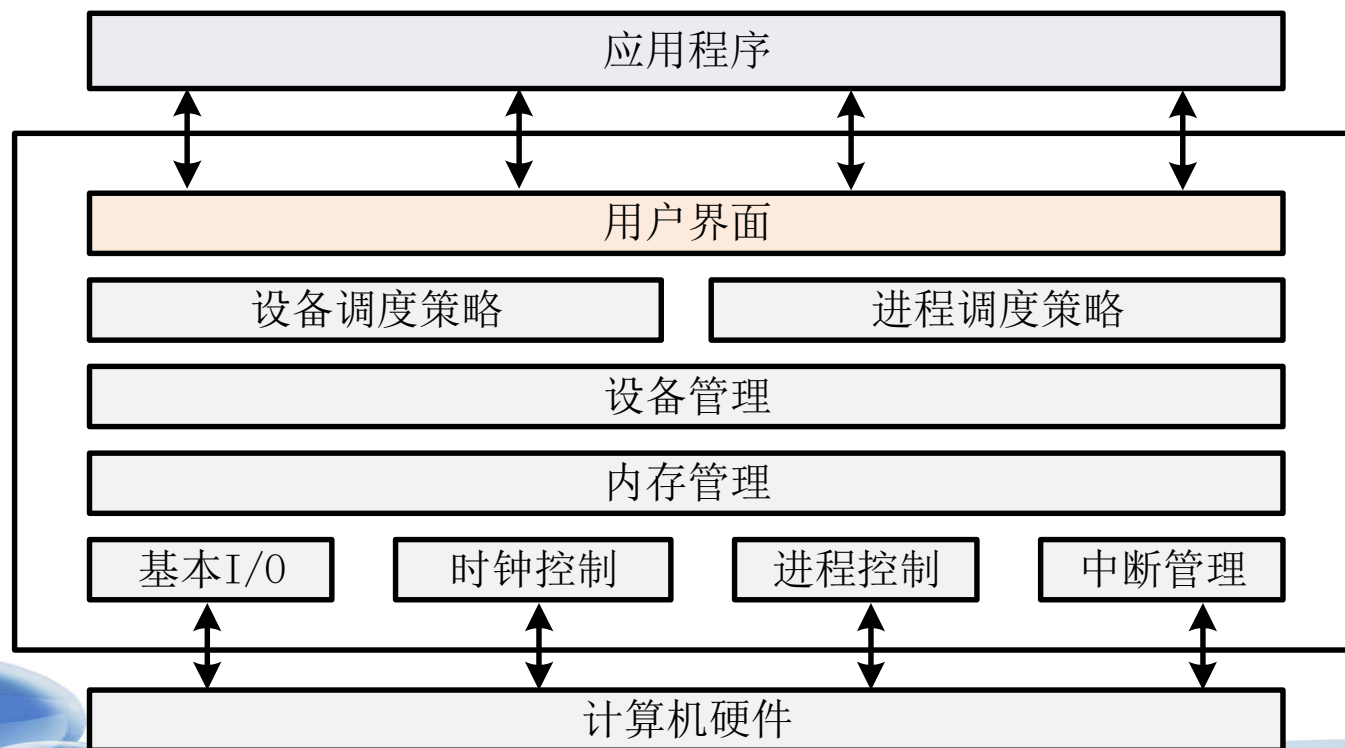
- 功能模块按调用次序排若干层，各层单向依赖或单向调用。



分层结构的OS

● 分层原则

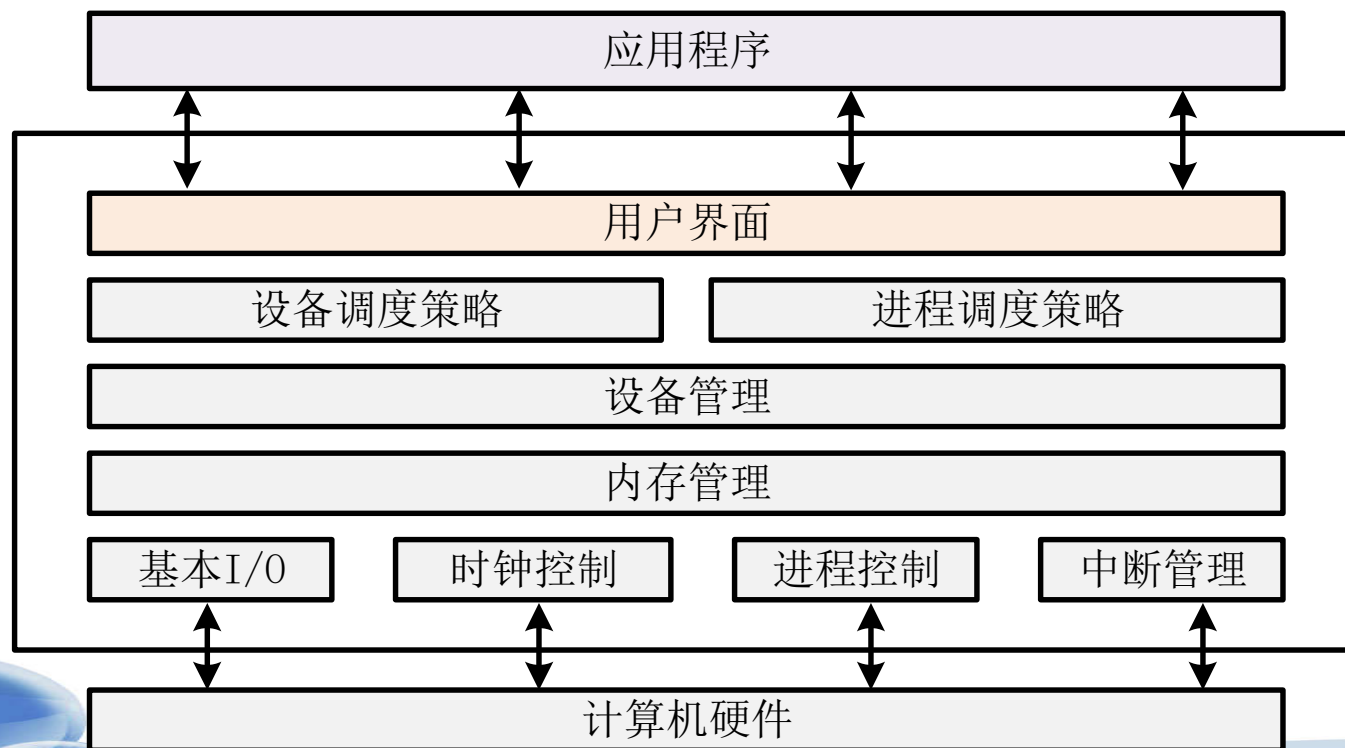
- (1) 最底层：硬件相关
- (2) 最顶层：用户策略/用户交互
- (3) 中间层：按调用次序/消息传递顺序
- (4) 较低层：共性的、活跃的服务



分层结构的OS

● 层次结构的优点

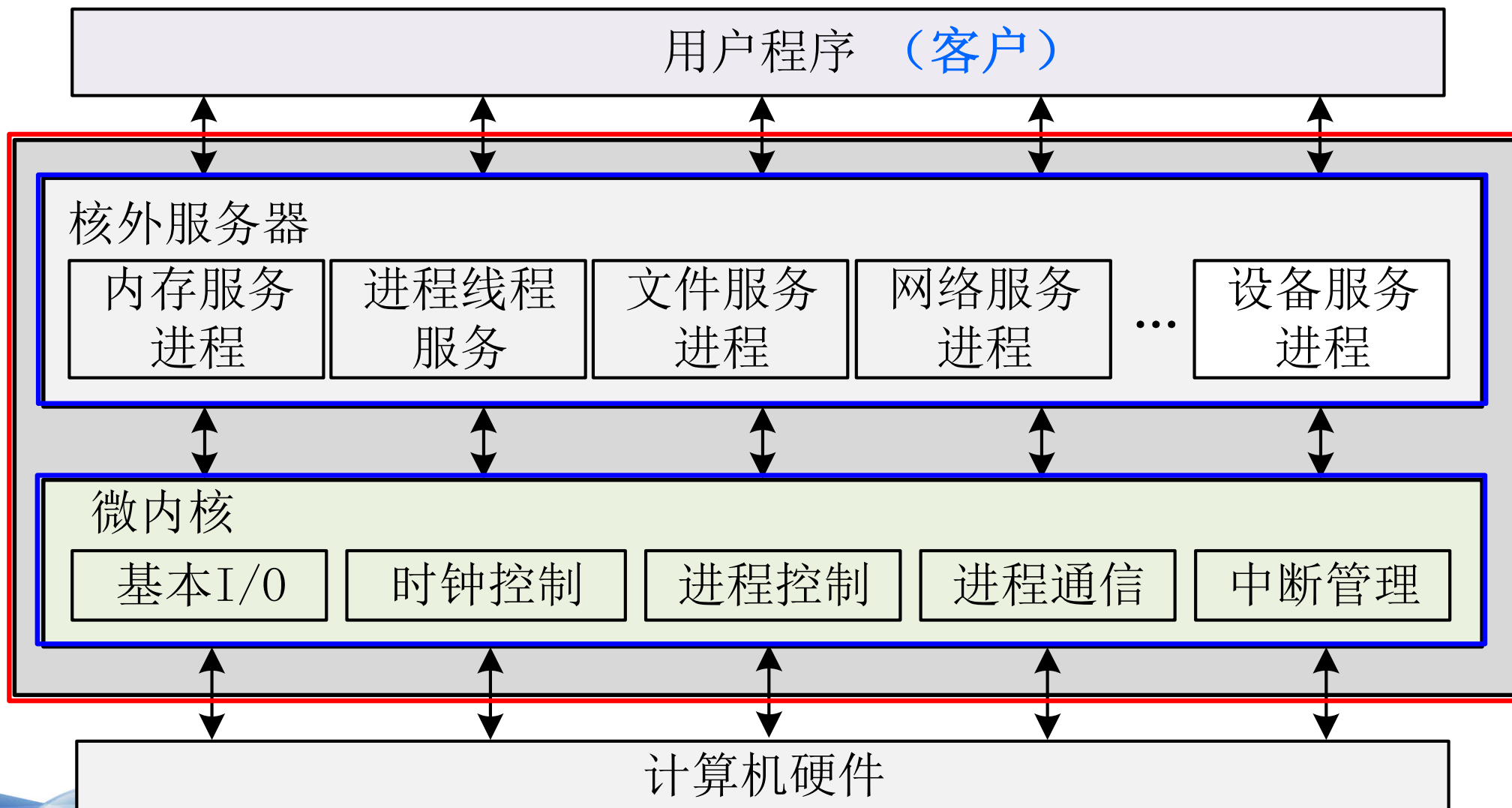
- 结构清晰，避免循环调用。
- 整体问题局部化，系统的正确性容易保证。
- 有利于操作系统的移植、维护、扩充。



(3) 微内核结构（客户/服务器结构）

- 操作系统 = 微内核 + 核外服务器

操作系统



(3) 微内核结构（客户/服务器结构）

- 客户：应用程序

- 服务器：操作系统 = 微内核 + 核外服务器

- 微内核

- ◆ 足够小，提供OS最基本的核心功能和服务

- 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 负责客户和服务器间的通信。

- 核外服务器

- ◆ 完成OS绝大部分功能，等待客户提出请求。

- ◆ 由若干服务器或进程共同构成

- 例如：进程/线程服务器，虚存服务器，设备管理服务器等，以进程形式运行在用户态。

(3) 微内核结构（客户/服务器结构）

- 典型的微内核操作系统

- Minix

- HarmonyOS

- RT-Thread

- FreeRTOS

- ARTs-OS

- Win NT

- COSIX

- 宏内核

- Macro-Kernel

- ◆ Linux

● 微内核和单体内核（宏内核）比较

类别	特点	优点	缺点	代表
单体内核 /宏内核	图形、设备驱动、文件系统与内核运行在同一地址空间	进程通信、状态切换效率高	内核庞大，不易剪裁；系统稳定性和安全性差。	UNIX Linux
微内核	内核只实现基本功能	内核精练、便于剪裁和移植；系统稳定性和安全性高。	进程状态频繁切换，系统效率低。	Minix WinNT