现代操作系统

(Modern Operating System)

福州大学　计数学院　江兰帆

课程特点

- 难教：概念多、抽象、枯燥
- 难学：理论性强、难理解
- 难考：记忆内容多、题目灵活

学习要求

按时上课，认真听讲，课外阅读。

学习方式与方法

比较学习法

- 不同设计思想与实现方法之间的差别

实践式学习法

- 脑过千遍不如手过一遍，坐着想不如动手实践

课程考察与考试

课堂作业

- 随原理讲授课程随机进行

成绩分布

- 笔试 + 实验 + 课堂表现

学术诚信

- 有些行为可能可以使你得到分数，但失去应有的训练

课件、资料下载

- 福州大学课程中心
  - http://met2.fzu.edu.cn

为什么学习操作系统

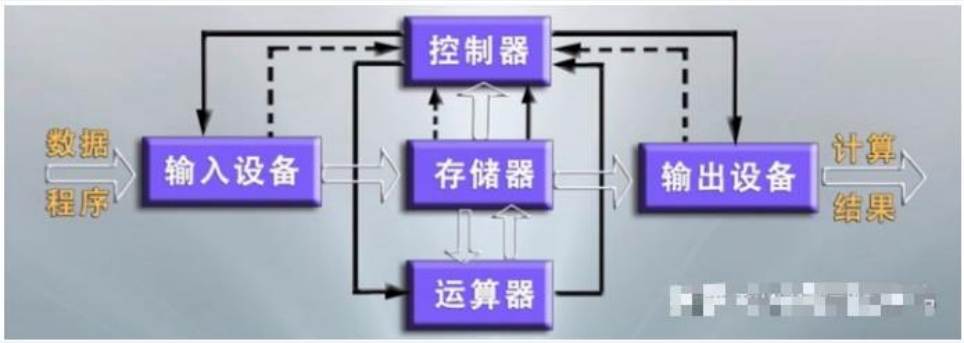
- 加深对使用的操作系统的理解，有利于深入编程
  - 用于为了开发应用程序必须与操作系统打交道。
- 编程时借鉴操作系统的设计思想和算法
  - 操作系统中许多概念和技巧可以推广应用到其他领域
- 设计操作系统 或 修改现有操作系统
  - “操作系统”移植应用于嵌入式系统

第1章 引论

- 1.1 什么是操作系统
- 1.2 操作系统的历史
- 1.3 计算机硬件介绍
- 1.4 操作系统大观
- 1.5 操作系统概念
- 1.6 系统调用
- 1.7 操作系统结构

1.1 什么是操作系统

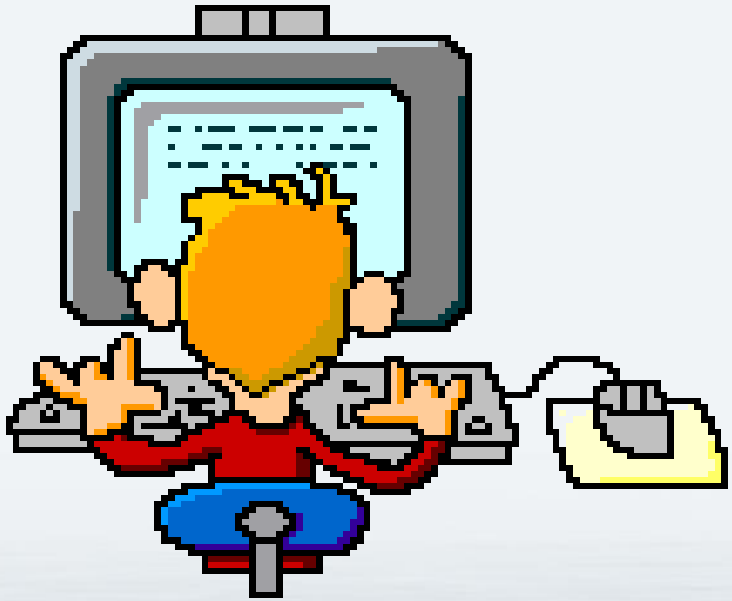
计算机系统组成



- 硬件
  - 计算机系统的物质基础
  - 运算器、控制器、存储器、输入/输出设备(裸机)
- 软件
  - 由硬件执行以完成一定任务的程序及其数据
  - 系统软件、应用软件

问题的提出

- 什么是计算机操作系统?
- 计算机操作系统有什么作用?

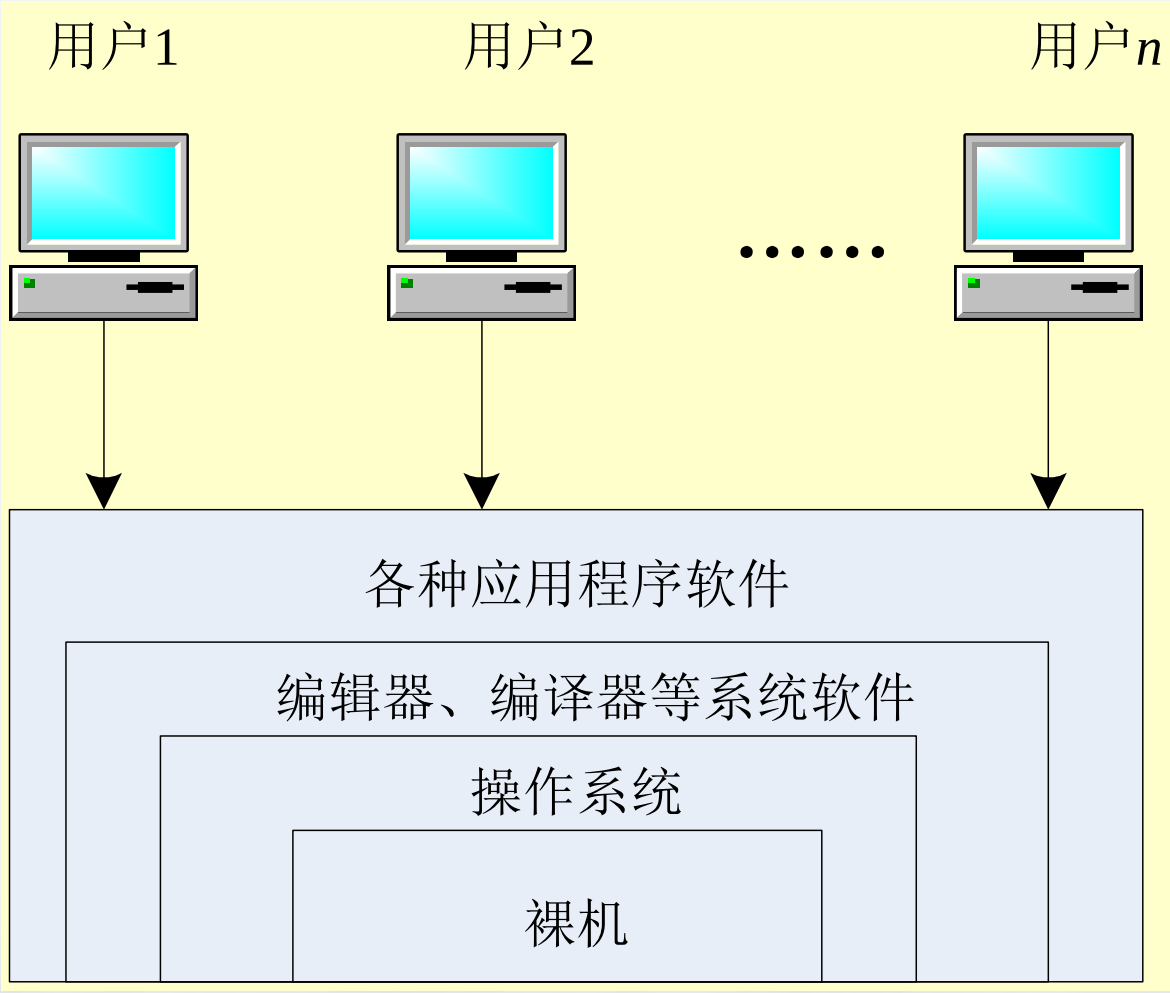


举例分析

我们以网易云音乐听歌为例：

- step1: 找到安装目录
- step2: 双击cloudmusic.exe
- step3: 网易云音乐运行中
- step4: 放歌

操作系统在软硬件层次中的地位



Operating System (OS)

- A body of software, in fact, that is responsible for *making it easy to run programs* (even allowing you to seemingly run many at the same time), allowing programs to share memory, enabling programs to interact with devices, and other fun stuff like that.

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，它直接控制和管理计算机系统中的硬件及软件资源，合理的组织计算机工作流程，以便有效的利用这些资源为用户提供一个功能强大、使用方便和可扩展的工作环境，从而在计算机与其用户之间起到接口的作用。

操作系统的定义

- 从硬件扩充的角度，操作系统是计算机裸机之上的第一层软件，它掩盖了硬件操作的细节，是对计算机硬件功能的第一次扩充。
- 从资源管理的角度，操作系统是控制和管理计算机硬、软件资源，合理地组织计算机的工作流程以及方便用户的程序集合。

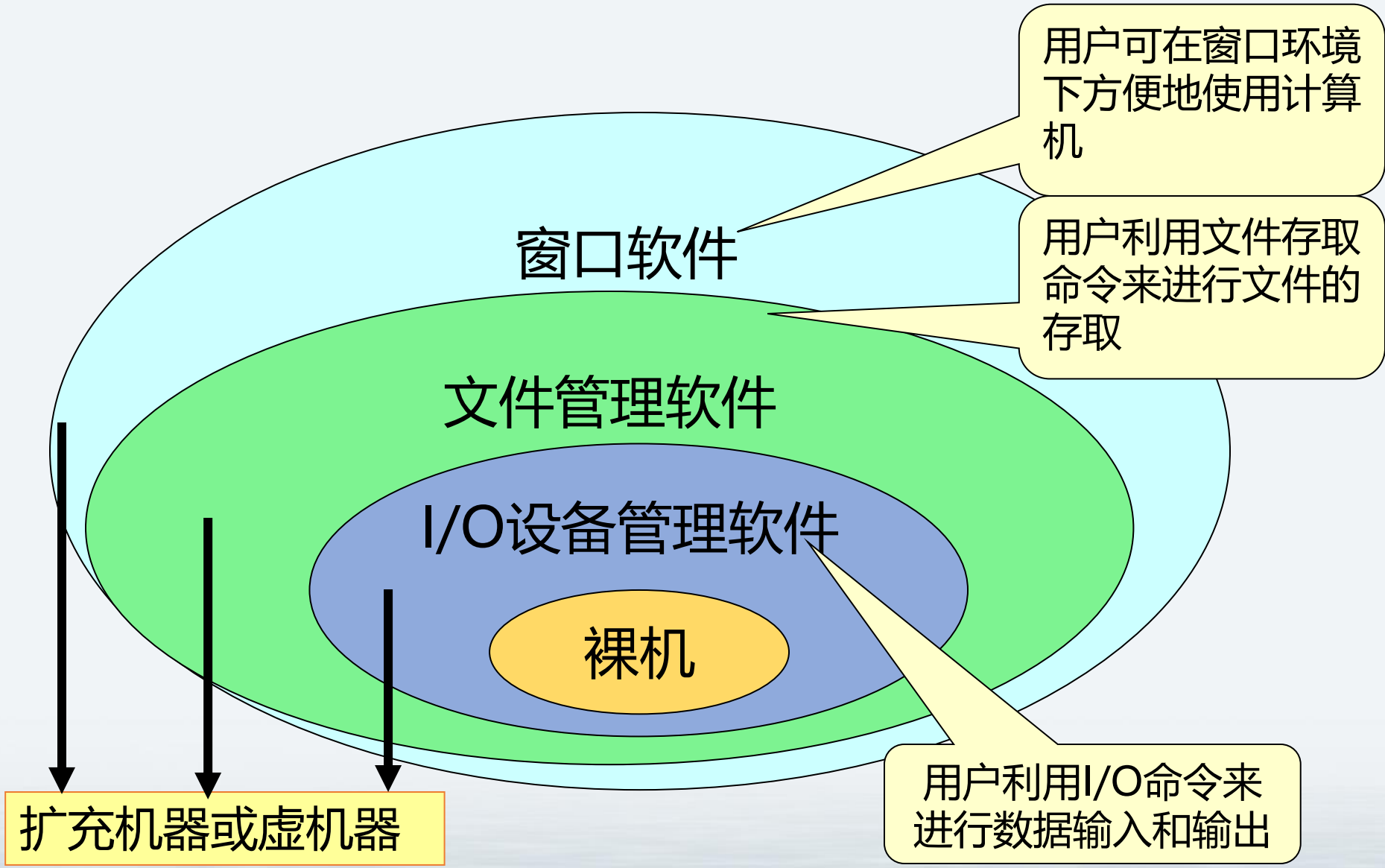
问题的提出

- 操作系统作为 系统资源的管理者 （资源包括硬件、软件、文件等），需要提供什么功能呢?
- 操作系统作为 用户与计算机硬件之间的接口，要为其上层的用户、应用程序提供简单易用的服务，需要提供什么功能?
- 操作系统作为最接近硬件的层次，需要在纯硬件的基础上实现什么功能?

1.1.1 作为扩展机器的操作系统

- 在没有操作系统的情况下，如何读取一个文件?
- 盘片号？磁道号？扇区号？……

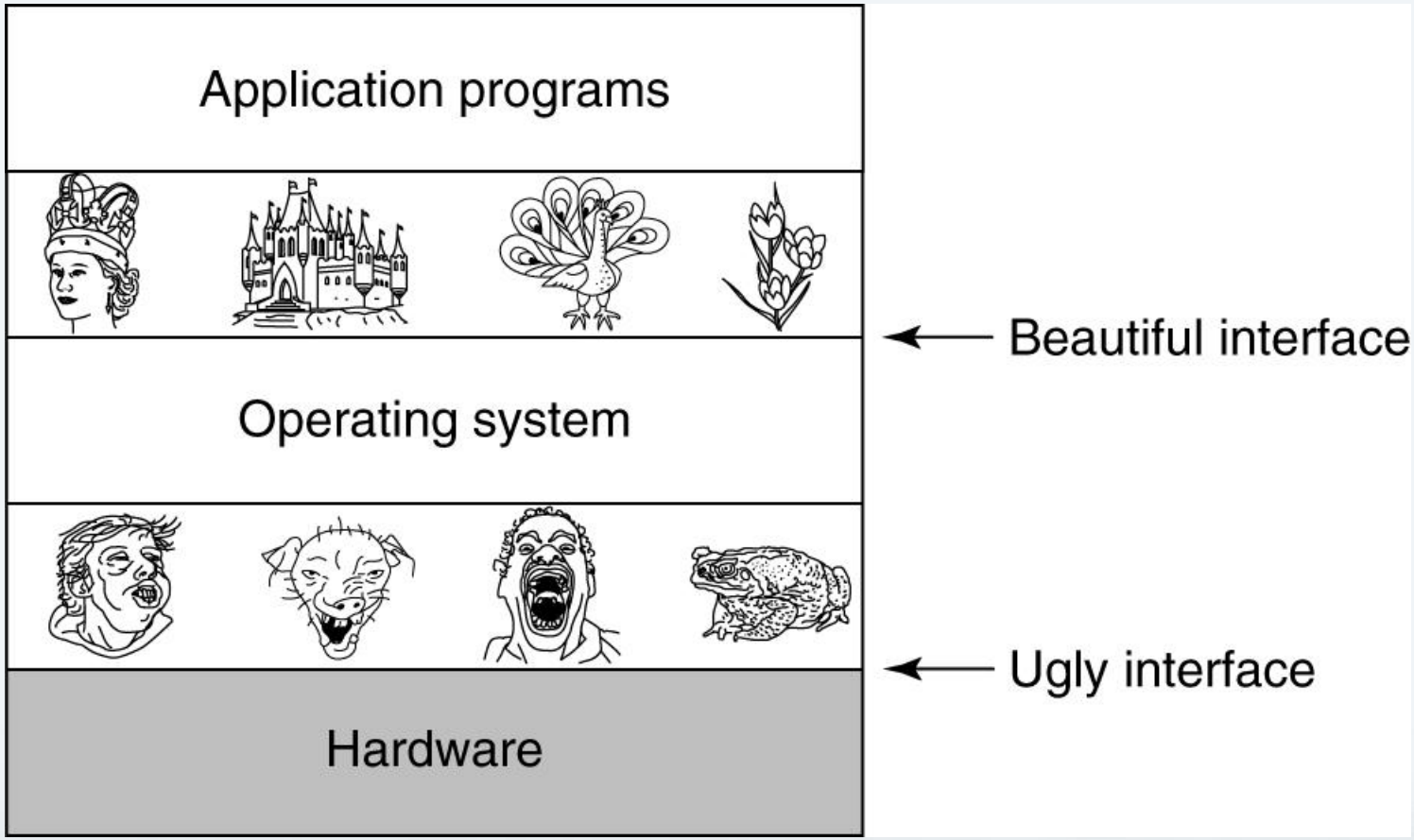
作为扩展机器的操作系统



1.1.1 作为扩展机器的操作系统

- 在裸机上覆盖OS后，便可获得一台功能显著增强，使用极为方便的扩充机器或虚拟机。
- One of the major tasks of the operating system is to hide the hardware and present programs (and their programmers) with nice, clean, elegant, consistent, abstractions to work with instead.

小结


Application programs
Operating system
Hardware
Beautiful interface
Ugly interface

1.1.2 作为资源管理者的操作系统

- 问题的提出：三个程序同时试图在一台打印机上输出计算结果？？？
- 操作系统的任务：在相互竞争的程序间有序地控制设备的分配。

1.1.2 作为资源管理者的操作系统

- 处理机管理， 用于分配和控制处理机；
- 存储器管理，主要负责内存的分配与回收；
- I/O设备管理，负责I/O设备的分配与操纵；
- 文件管理，负责文件的存取、共享和保护。

1.1.2 作为资源管理者的操作系统

资源管理的实现方式：

- 时间复用：CPU的分时共享
- 空间复用：内存的同时共享

小结

- 对计算机系统而言，操作系统是对所有系统资源进行管理的程序的集合；
- 对用户而言，操作系统提供了对系统资源进行有效利用的简单抽象的方法。

1.2 操作系统的历史

1.2 操作系统的历史

- 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 操作系统的历史

操作系统的发展和计算机的组成与体系结构相关，经历了四个发展阶段：

① 1946年--50年代末：电子管时代，无操作系统。

② 1950年代末--60年代中期：晶体管时代，批处理系统。

1.2 操作系统的历史

③ 1960年代中期--70年代中期：集成电路时代，多道程序设计。

④ 1970年代中期至今：大规模和超大规模集成电路时代，分时、实时系统。 现代计算机正向着巨型、微型、并行、分布、网络化和智能化几个方面发展。

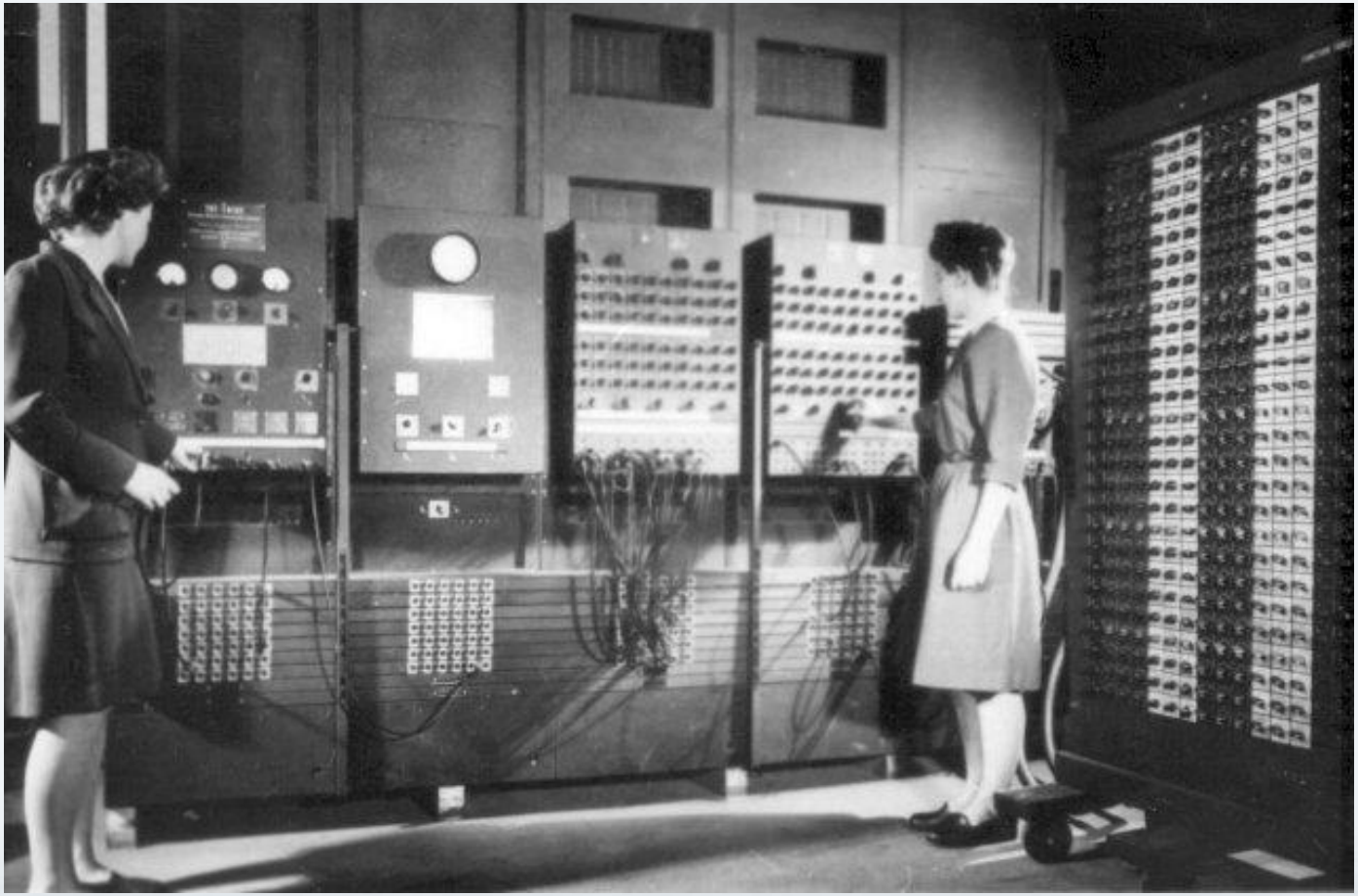
1.2.1 无操作系统的计算机系统

- 时间：1946 ~ 50年代末
- 主要器件工艺：电子管
- 运算速度：慢，1000次/秒
- 没有操作系统
- 程序设计语言：机器语言

手工操作阶段—— ENIAC(1946.2.14)

1945年，美国宾夕法尼亚大学莫尔学院

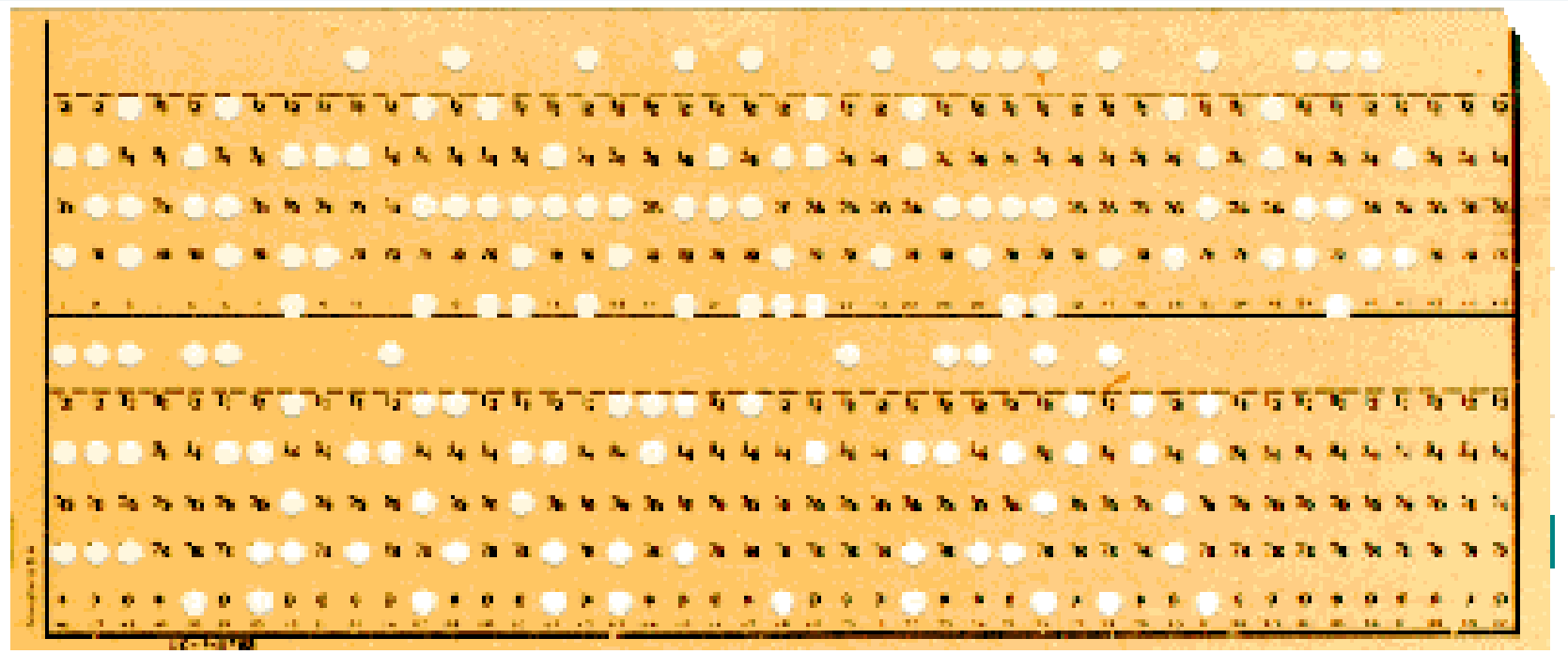
两位女士正在操作ENIAC



无操作系统的计算机系统

- 工作量大，难度高，易出错，需要大量人力和物力；
- 用户：用户既是程序员，又是操作员；用户是计算机专业人员；
- 编程语言：机器语言；
- 输入输出：纸带或卡片；

输入输出：纸带或卡片



人工操作方式

- 程序员将程序写在卡片上（在卡片上穿孔）。
- 程序员先预约，然后到机房将他的卡片放入卡片输入机，启动输入机将卡片上的程序和数据读入计算机。
- 打开控制台开关，启动程序运行，打印机输出计算结果。
- 程序员卸下卡片，下一个程序员上机。

1.2.1 无操作系统的计算机系统

- 缺点：
  - 用户独占全机，资源利用率低
  - CPU等待人工操作。
- 解决办法：
  - 设立专门的操作员：减少操作错误；
  - 批处理——实现作业的自动过渡

1.2.2 单道批处理系统

- 时间：50年代末 ~ 60年代中
- 主要器件工艺：晶体管
- 运算速度：几十万至百万次/秒，I/O 设备的速度已经严重低于处理器的速度
- 操作系统：监督程序——早期操作系统雏形
- 程序设计语言——汇编语言和高级语言(如FORTRAN)

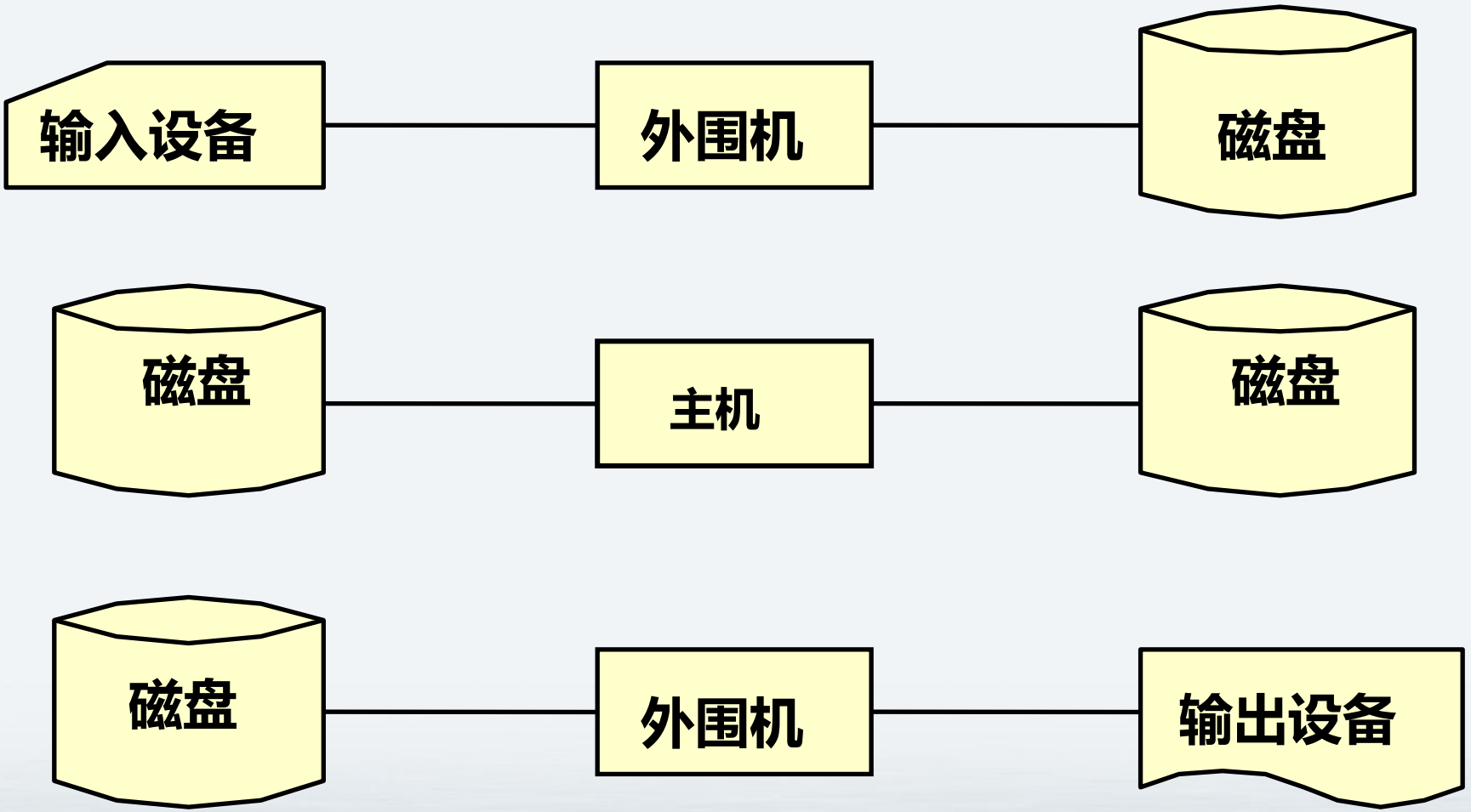
1.2.2 单道批处理系统

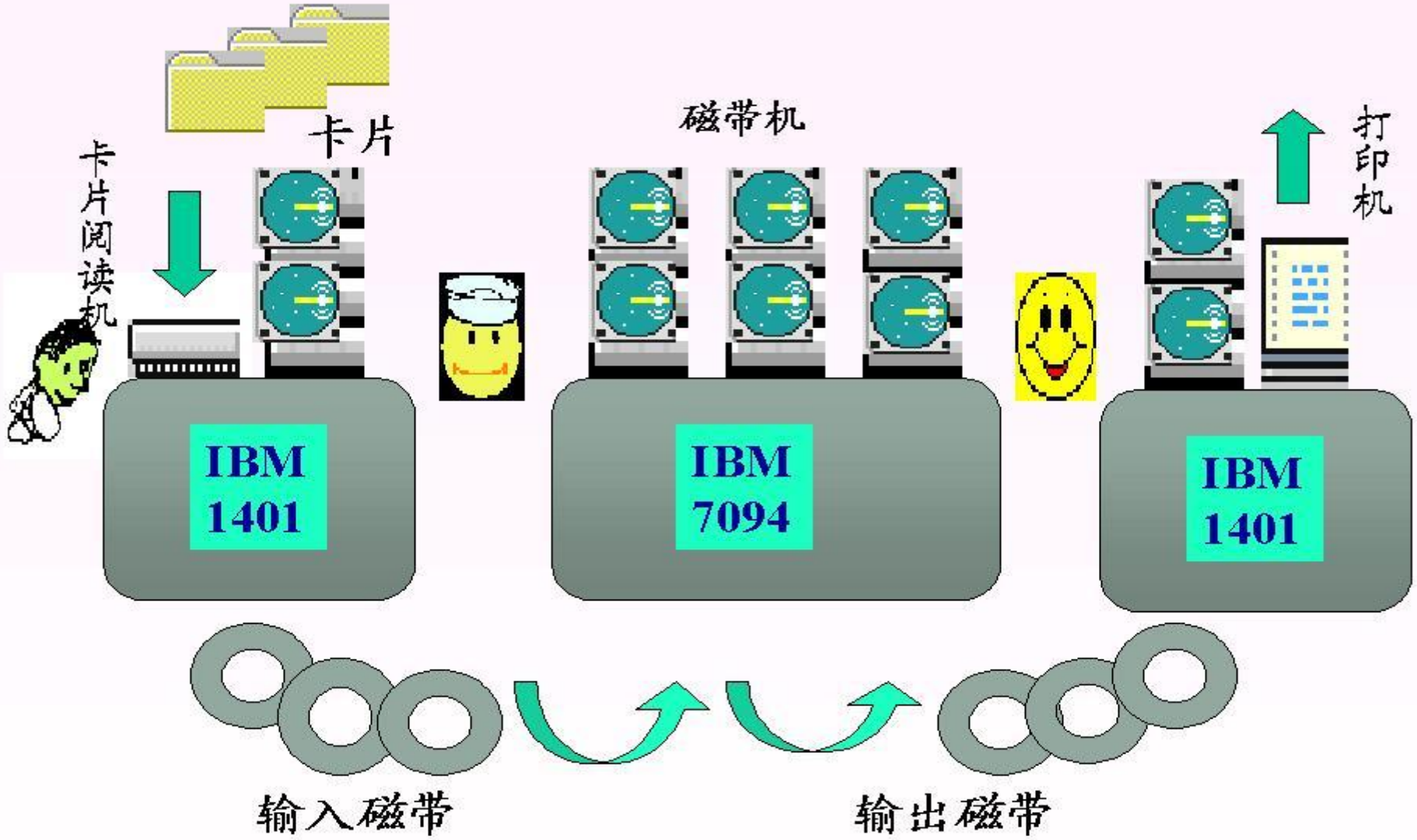
- 所谓批处理系统是指加载在计算机上的一个系统软件，在它的控制下，计算机能够自动、成批地处理一个或多个用户的作业。
- 单道是指在内存中始终只保持一道作业。

1.2.2 单道批处理系统

- 在主机之外另设一台外围计算机，它不与主机直接连接，只与外部设备打交道
  - 外围机把读卡机上的作业逐个地传送到输入磁带机上
  - 主机只负责把作业从磁带上调入内存并运行它，作业完成后主机把结果记录到输出磁带上
  - 外围机负责把输出磁带上的信息读出来，并交打印机打印

1.2.2 单道批处理系统




卡片
卡片阅读机
磁带机
打印机
IBM
1401
IBM
7094
IBM
1401
输入磁带
输出磁带

1.2.2 单道批处理系统

优点:

- 减少了CPU的空闲时间;
- 提高I/O速度。

系统利用率

从文件中读取一条记录	0.0015秒
执行100条处理指令	0.0001秒
向文件中写回该记录	0.0015秒
总计	0.0031秒
利用率	3.2%

- 单道程序设计

问题的提出

CPU和I/O设备使用不均衡（取决于当前作业的特性）

- 对计算为主的作业，外设空闲；
- 对I/O为主的作业，CPU空闲；

如何使CPU和外设都能尽量保持忙碌？

1.2.3 多道批处理系统

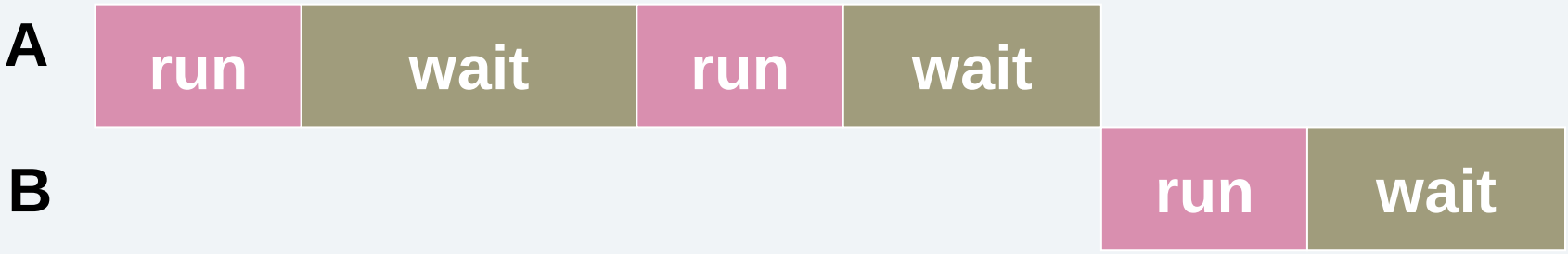
- 在单道批处理系统中，内存中仅有一道作业，这使系统中仍有较多的空闲资源，致使系统的性能较差。
- 为了进一步提高资源的利用率和系统对作业的吞吐量，在20世纪60年代中期，引入了多道程序设计技术，由此而形成了多道批处理系统。

1.2.3 多道批处理系统

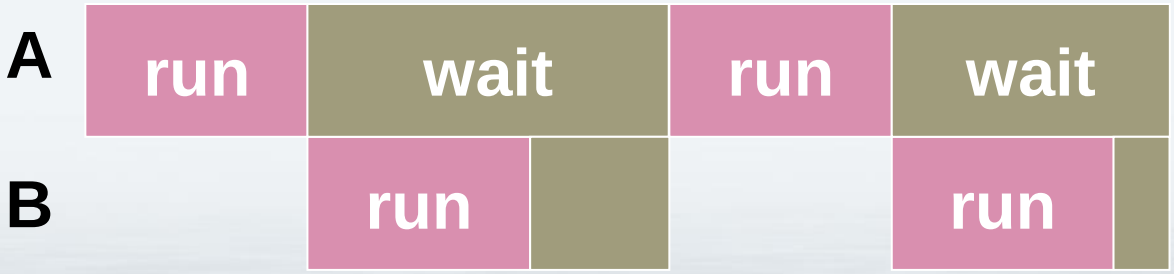
- 多道程序设计技术就是在内存中同时保持若干道程序，系统按某种调度策略交替执行这些程序，使CPU保持最少的空闲时间。
- 多道程序设计的主要优点是通过将用户的CPU请求和I/O请求重叠起来的办法来有效地使用CPU。它设法让CPU总有事情可做，以此来提高CPU的利用率。

多道程序设计 multiprogramming

- 单道程序设计



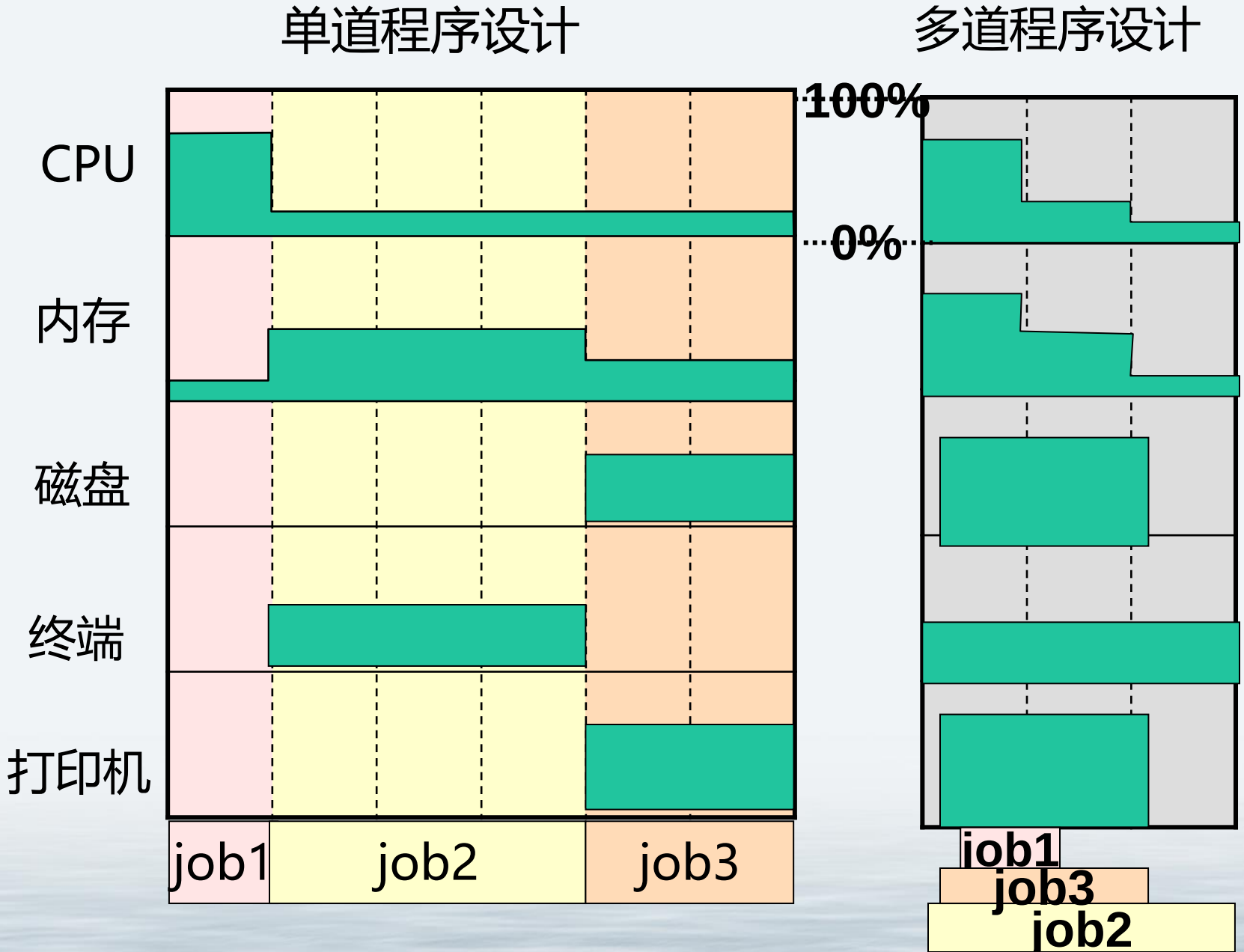
- 多道程序设计



假设系统中有三个作业…

	JOB1	JOB2	JOB3
作业类型	大量计算	大量I/O	大量I/O
持续时间	5min	15min	10min
作业大小	50k	100k	80k
需要磁盘	否	否	是
需要终端	否	是	否
需要打印机	否	否	是

执行情况对比


单道程序设计
多道程序设计
CPU
内存
磁盘
终端
打印机
100%
0%
job1
job2
job3
job1
job3
job2

资源利用情况

	单道程序	多道程序
CPU	22%	43%
内存	30%	67%
磁盘	33%	67%
打印机	33%	67%
总时间	30min	15min
吞吐率	6个/h	12个/h

1.2.3 多道批处理系统

多道程序的运行特点：

- 多道：计算机内存中同时存放多道相互独立的程序。
- 宏观上并行：同时进入系统的几道程序都处于运行状态，但都未运行完。
- 微观上串行：各作业交替使用CPU，交替执行。

1.2.3 多道批处理系统

多道批处理系统需要解决的问题：

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

1.2.3 多道批处理系统

多道批处理系统的优缺点：

- 资源利用率高
- 系统吞吐量大
- 作业平均周转时间长
- 无交互能力

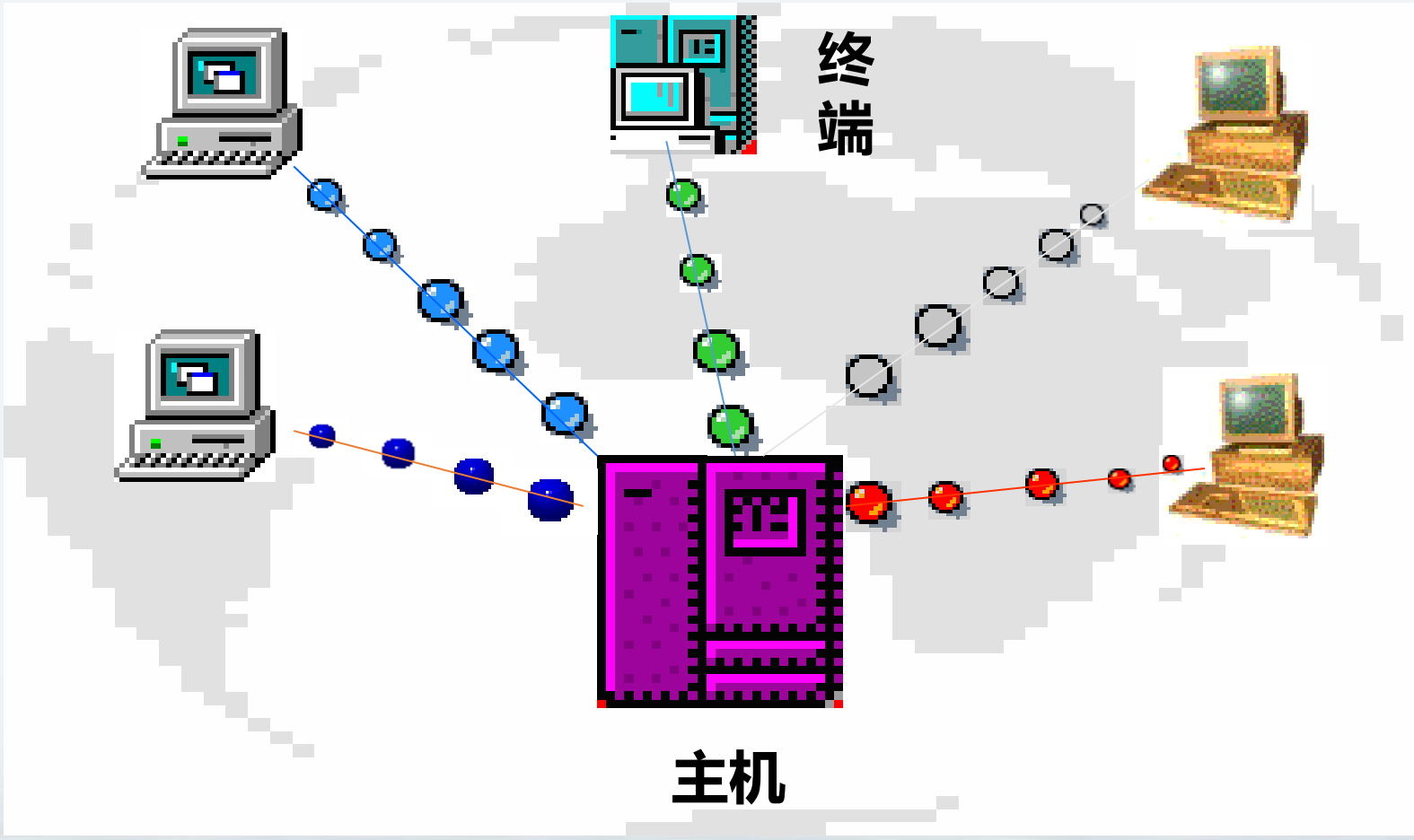
1.2.4 分时系统

- 多道批处理系统：提高资源利用率和系统吞吐量
- 分时系统：满足用户需求,如良好的人机交互性

1.2.4 分时系统

- 分时系统实现中最关键的问题是如何使用户能与自己的作业进行交互，即当用户在自己的终端上键入命令时，系统应能及时接收并及时处理该命令，再将结果返回给用户。应强调指出，即使有多个用户同时通过自己的键盘键入命令，系统也应能全部地及时接收并处理。

1.2.4 分时系统



1.2.4 分时系统

- 分时技术是把处理机的时间分成很短的时间片，这些时间片轮流地分配给联机的各作业使用。
- 如果某作业在分配给它的时间片用完时仍未完成，则该作业就暂时中断，等待下一轮运行，并把处理机的控制权让给另一个作业。
- 在一个相对较短的时间间隔内，每个用户作业都能得到快速响应，以实现人机交互。

1.2.4 分时系统

- 采用分时技术的系统称为分时系统(Time-Sharing System)。在分时系统中，一台主机上连接多个带有显示器和键盘的终端，同时允许多个用户通过自己的键盘，以交互的方式使用计算机，共享主机中的资源。

分时系统主要特征

- 多路性：允许在一台主机上同时联接多台联机终端，系统按分时原则为每个用户服务。
- 独立性：每个用户各占一个终端，彼此独立操作，互不干扰。
- 及时性：用户的请求能在很短时间内获得响应。
- 交互性：用户可通过终端与系统进行广泛的人机对话。

1.2.5 实时系统

- 实时系统(Real-Time System)是指系统能及时(或即时)响应外部事件的请求，在规定的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。

1.2.5 实时系统

- 实时系统是在响应时间方面有严格制约的专用系统。实时系统与分时系统的区别在于：在分时系统中，快速响应是需要的，但不是必需的；在实时系统中，处理事务必须在适合于此系统的特定时间限额内完成。

截止时间
(Deadline)

1.2.5 实时系统

根据对截止时间的要求来划分：

(1) 硬实时任务(hard real-time task)。系统必须满足任务对截止时间的要求，否则可能出现难以预测的结果。

(2) 软实时任务(Soft real-time task)。它也联系着一个截止时间， 但并不严格，若偶尔错过了任务的截止时间，对系统产生的影响也不会太大。

1.2.5 实时系统

实时系统有两类典型的应用形式：

- 实时控制系统：通常是指以计算机为中心的生产过程控制系统。
- 实时信息处理系统：通常是指对信息进行实时处理的系统，如飞机订票系统、情报检索系统等。

1.2.5 实时系统

实时操作系统主要是为联机实时任务服务的，相比分时系统它有其自身的特点：

① 与分时系统一样具有多路性和独立性。

② 及时响应：对外部实时信号必须能及时响应，响应的时间间隔要足以控制发出实时信号的那个环境。

1.2.5 实时系统

实时操作系统主要是为联机实时任务服务的，相比分时系统它有其自身的特点：

③ 高可靠性和安全性，系统的效率则放在第二位。

④ 简单交互性。一般仅是针对待定的实时任务提供一些简短的操作命令。

小结

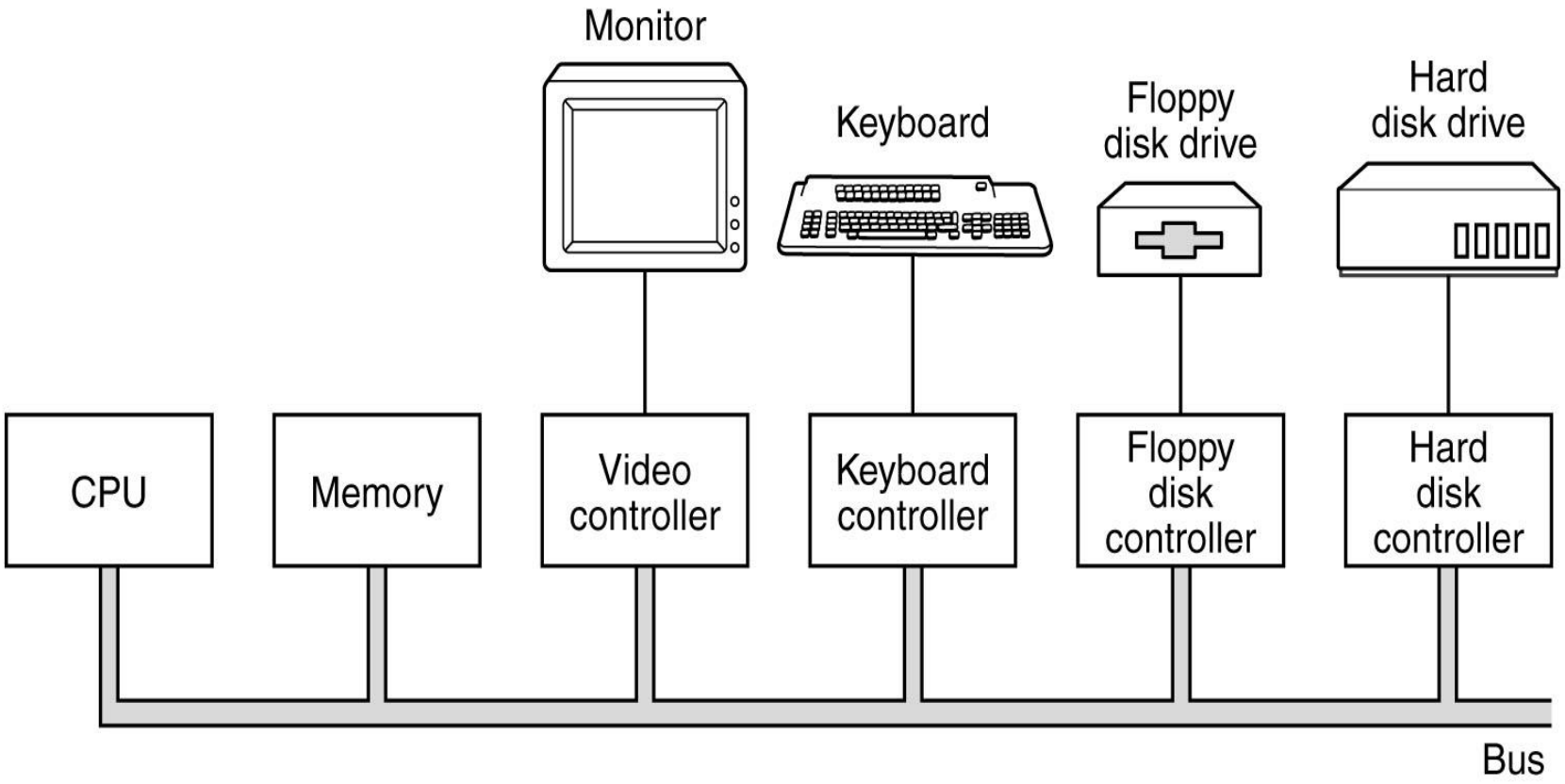
- 操作系统是在人们使用计算机的过程中，为了满足两大需求：提高资源利用率、增强计算机系统性能，伴随着计算机技术本身及其应用的日益发展，而逐步地形成和完善起来的。

当代操作系统的两大发展方向

- 宏观应用：如分布式操作系统、机群操作系统、WebOS、云操作系统等。
- 微观应用：如嵌入式操作系统。

1.3 计算机硬件介绍

1.3 计算机硬件介绍


Monitor
Keyboard
Floppy disk drive
Hard disk drive
CPU
Memory
Video controller
Keyboard controller
Floppy disk controller
Hard disk controller
Bus

1.4 操作系统大观

1.4 操作系统大观

- 大型机操作系统
- 服务器操作系统
- 多处理机操作系统
- 个人计算机操作系统
- 实时操作系统
- 嵌入式操作系统

1.5 操作系统中的主要概念

1.5 操作系统中的主要概念

- 进程
- 地址空间
- 文件
- 输入/输出
- 保护
- shell

概念的重用

技术的变化会导致某些思想过时并迅速消失，但是，技术的另一种变化还可能再次复活某些思想。如：

组合逻辑控制方式

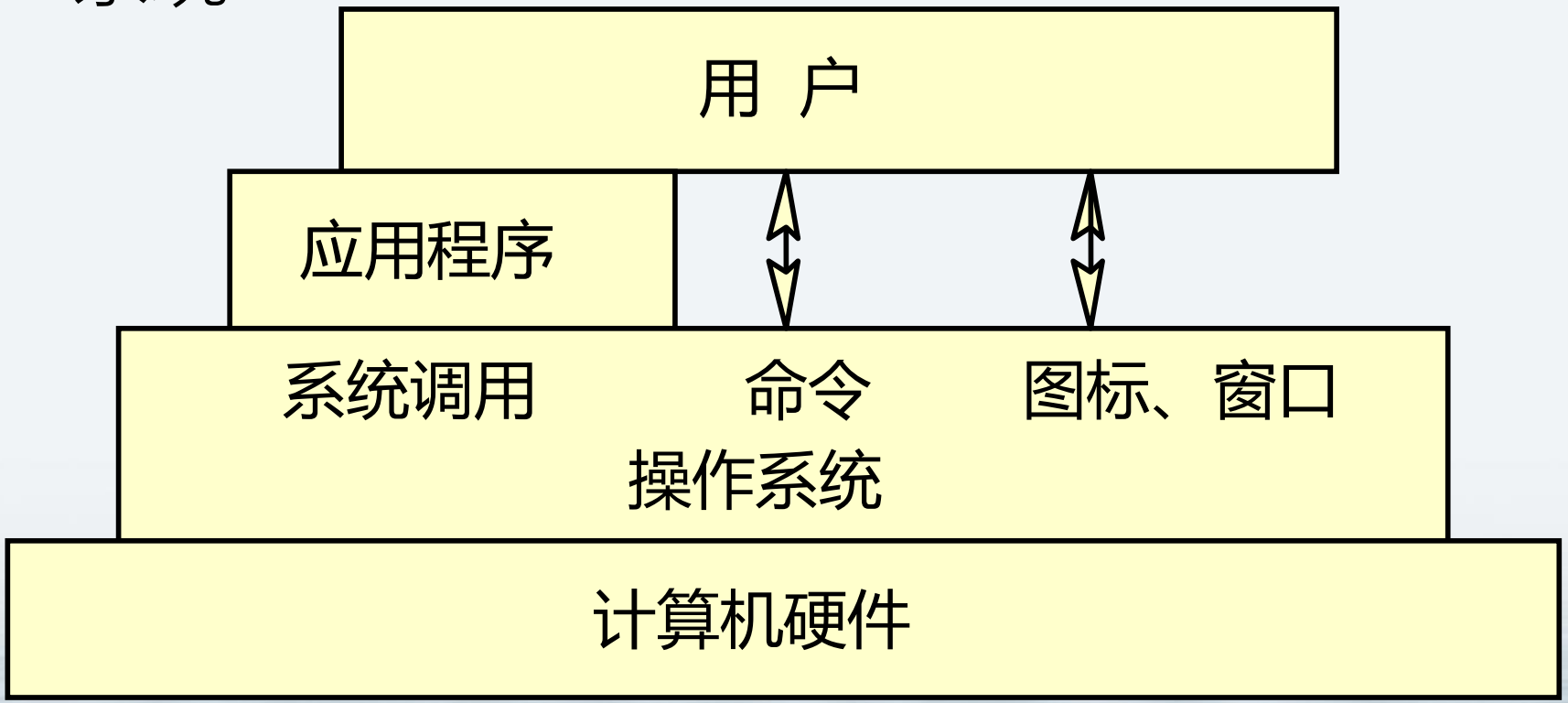
→微程序控制方式

→组合逻辑控制方式

1.6 系统调用

1.6 系统调用

- Q: 我们作为用户，到底怎么“使用”操作系统?



1.6 系统调用

计算机系统中有两种状态：

- 核心态：操作系统内核程序执行时所处的状态。这种状态具有较高的特权，能执行一切指令，访问所有的寄存器和存储区。
- 用户态：用户程序运行时所处的状态。这种状态具有较低的特权，只能执行规定的指令，访问指定的寄存器和存储区。

1.6 系统调用

- 例如，我们知道Linux的运行空间分为内核空间与用户空间，它们各自运行在不同的级别中，逻辑上相互隔离。所以用户进程在通常情况下不允许访问内核数据，也无法使用内核函数，它们只能在用户空间操作用户数据，调用用户空间函数。

1.6 系统调用

- 系统服务之所以需要通过系统调用来提供给用户空间的根本原因是为了对系统进行“保护”。

1.6 系统调用

- 系统调用，顾名思义，说的是操作系统提供给用户程序调用的一组“特殊”接口。用户程序可以通过这组“特殊”接口来获得操作系统内核提供的服务，比如用户可以通过文件系统相关的调用请求系统打开文件、关闭文件或读写文件，可以通过时钟相关的系统调用获得系统时间或设置定时器等。

1.6 系统调用

- 从逻辑上来说，系统调用可被看成是一个内核与用户空间程序交互的接口。它好比一个中间人，把用户进程的请求传达给内核，待内核把请求处理完毕后再将处理结果送回给用户空间。

1.6 系统调用

- 系统调用的特殊性主要在于规定了用户进程进入内核的具体位置；换句话说，用户访问内核的路径是事先规定好的，只能从规定位置进入内核，而不准许肆意跳入内核。有了这样的陷入内核的统一访问路径限制才能保证内核安全无虞。

- Shell 是一门 “把用户指令翻译成系统调用” 的编程语言（man sh）

1.7 操作系统结构

1.7 操作系统结构

从操作系统的发展来看,采用的体系结构一般有4种:

- 整体结构
- 分层结构
- 虚拟机结构
- C/ S结构(微内核结构)

整体式结构设计

- 它常被誉为“大杂烩”，这种结构方式下，开发人员为了构造最终的目标操作系统程序，首先将一些独立的过程，或包含过程的文件进行编译，然后用链接程序将它们链接成为一个单独的目标程序。

整体式结构设计

模块化结构设计

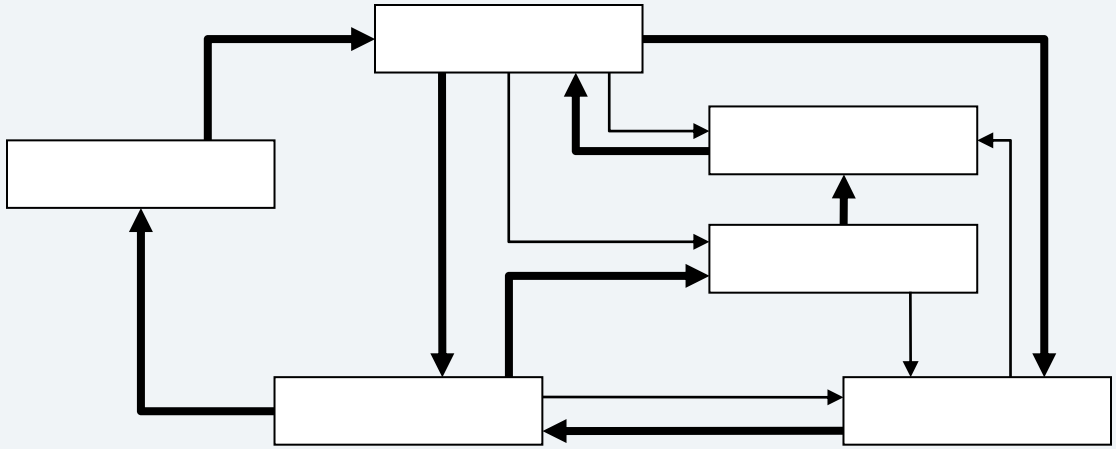
- OS按其功能划分为若干个具有一定独立性和大小的模块，每个模块具有某方面的管理功能。各模块间通过规定好的接口进行交互。
- 模块接口法是OS较早采用的一种结构程序设计方法，早期操作系统（IBM的OS）和小型OS（如MS-DOS）均属此类型。

整体式结构设计

- 模块的最大特点是将接口和其实现分离开来，这样就能够保证一个模块可以在不影响其他模块的情况下进行改变，增强了OS的可适应性。
- 加速了OS的开发过程。
- 增加了OS灵活性，便于修改和维护。但由于模块接口复杂，使得系统的结构关系不清晰，因而使系统的可靠性降低。

整体式结构设计

- 模块组合结构



- 为了减少各模块之间无序调动、互相依赖关系，特别是清除循环现象，引入层次结构设计法。

层次化结构设计

- 操作系统按一定的功能模块分层组织，最高层为用户程序，最底层为处理机调度及实现多道程序，并且下一层是相邻上一层的基础，层与层之间有严格的接口定义，只在相邻层之间发生交互。

层次化结构设计

5	操作员
4	用户程序
3	文件管理
2	设备管理
1	内存和磁盘管理
0	处理机分配和多道程序
	裸机

层次化结构设计

- 各层之间的模块只能是单向调用关系，即是只允许上层模块调用下层模块。
- 操作系统的结构清晰，而且不构成循环。
- 系统的调试和验证变得容易。
- 纯层次化模型实现的执行速度可能会较慢。

层次的设置

- 程序调用关系：被调用层应在调用层之下
- 程序运行频率：随着层次的增高，相应软件运行速度下降，因此，经常使用的模块应放在低层
- 公用模块：设置在最低层
- 用户接口：设置在最高层

C/S结构(微内核结构)

操作系统一般分为宏内核与微内核两种 。

- 宏内核（macrokernel）：也称单内核。整个系统是一个大模块，可以被分为若干逻辑模块，即处理器管理、存储器管理、设备管理和文件管理，其模块间的交互是通过直接调用其他模块中的函数实现的。

C/S结构(微内核结构)

- 微内核（microkernel）：把操作系统结构中的内存管理、设备管理、文件系统等高级服务功能尽可能地从内核中分离出来，变成几个独立的非内核模块，而在内核只保留少量最基本的功能，使内核变得简洁可靠，因此叫微内核。

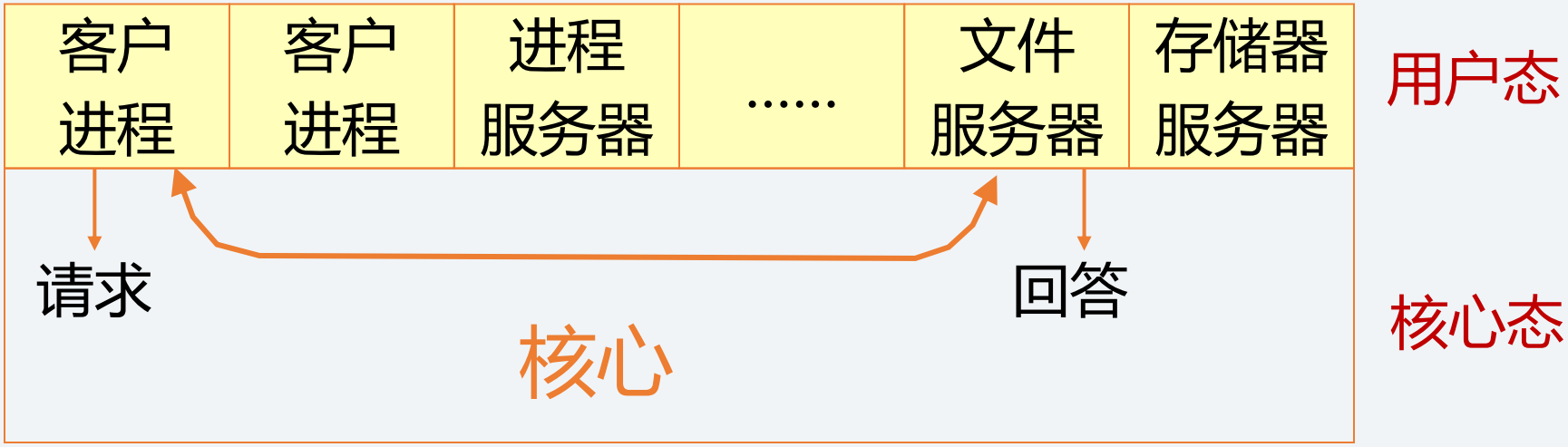
C/S结构(微内核结构)

C/S结构的思想是把OS分成两部分：

- 服务器进程：实现单个的一套服务（如：主存服务、进程生成服务、文件管理服务），运行在用户态。
- 内核：处理客户和服务器之间的通信。

C/S结构(微内核结构)

- 单机环境下的C/S模式



C/S结构(微内核结构)

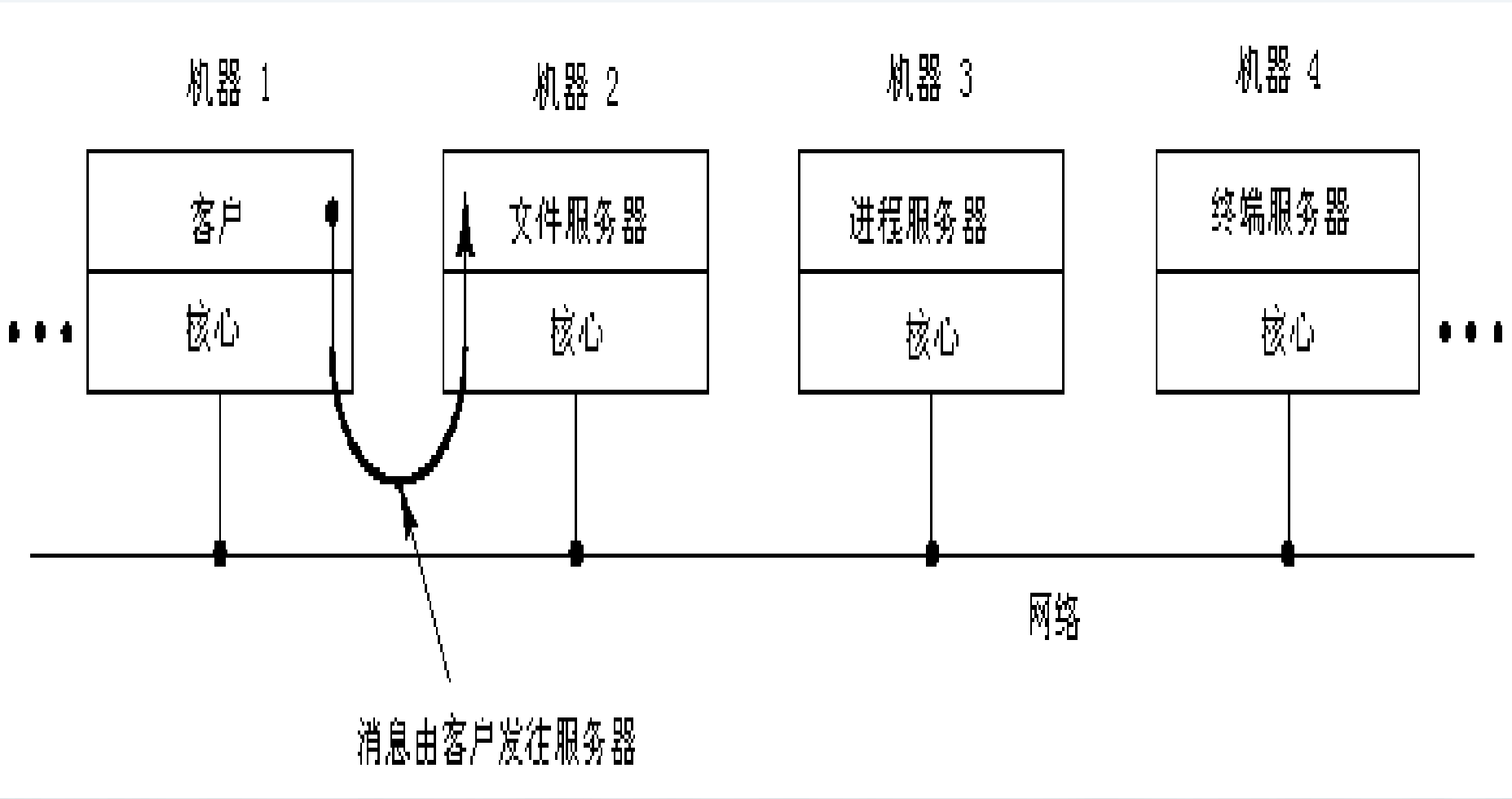
- 微内核部分经常只不过是一个消息转发站！

C/S结构(微内核结构)

C/S结构的优点:

- 良好的扩充性：只需添加支持新功能的服务进程即可。
- 可靠性好：服务器出错时，通常仅影响自身，不会导致整个系统瘫痪。
- 便于网络服务，实现分布式处理。

分布式系统中的客户/服务器模型



C/S结构(微内核结构)

- 充分的模块化，可独立更换任一模块而不会影响其他模块，从而方便第三方开发、设计模块。
- 未被使用的模块功能不必运行，因而能大幅度减少系统的内存需求。
- 具有很高的可移植性：只需要把微内核本身进行移植就可以完成将整个内核移植到新的平台上。其他模块都只依赖于微内核或其他模块，并不直接直接依赖硬件。

C/S结构(微内核结构)

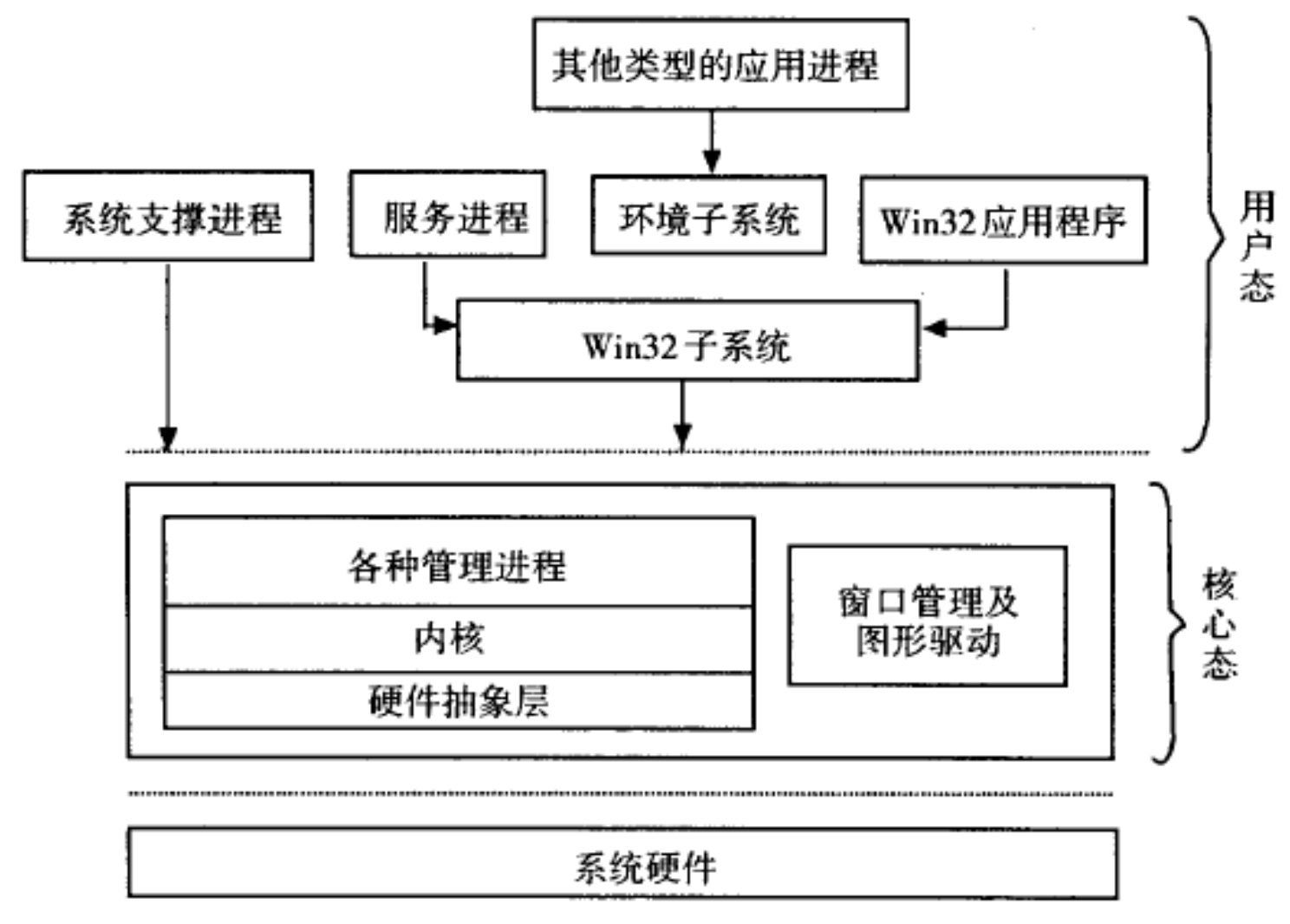
对比:

- 宏内核：执行效率较高
- 微内核：具有很好的可移植性，但效率较低(但可以通过提高硬件性能来补偿)

虚拟机结构

- 它以运行在裸机上的核心软件(虚拟机监控软件/或某一种操作系统)为基础，向上提供虚拟机的功能，每个虚拟机都像是裸机硬件的一个拷贝。在不同的虚拟机上可以安装不同的操作系统。

Windows 2000内核体系结构



本章小结

- Q1 (Why): 为什么要学操作系统?
- Q2 (What): 到底什么是操作系统?
  - 操作系统的历史
  - 操作系统结构

本章小结

- 计算机系统中的很多知识是关联的，对体系结构、编译器、软件工程等领域的理解都会加深对操作系统的理解；反之也一样。认识通常是“螺旋式上升”的；
- 计算机系统不是纸上谈兵，因此学习很多技术是非常重要的

课堂练习

用户要在程序获得系统帮助，必须通过（ ）。

A. 进程调度

B. 作业调度

C. 键盘命令

D. 系统调用

课堂练习

批处理系统的主要缺点是（ ）。

A. CPU的利用率不高

B. 失去了交互性

C. 不具备并行性

D. 以上都不是

课堂练习

在分时系统中，时间片一定时，（　），响应时间越长。

A. 内存越多

B. 用户数越多

C. 内存越少

D. 用户数越少

课堂练习

设计实时操作系统时，首先应考虑系统的（ ）。

A. 可靠性和灵活性

B. 实时性和可靠性

C. 灵活性和可靠性

D. 优良性和分配性

课堂练习

分时系统追求的目标是（ ）。

A. 充分利用I/O设备

B. 快速响应用户

C. 提高系统吞吐率

D. 充分利用内存

预习并思考

- 什么是进程?
- 为什么引入进程的概念?
- 进程与程序有什么不同?

THANKS

JIANG LANFAN

2022/8/28