



第1章 操作系统绪论



提纲

- 1.1 操作系统的概念
- 1.2 操作系统的形成与发展
- 1.3 操作系统的基本特征
- 1.4 操作系统的作用和功能
- 1.5 操作系统提供的服务和接口
- 1.6 操作系统的结构
- 1.7 典型的操作系统



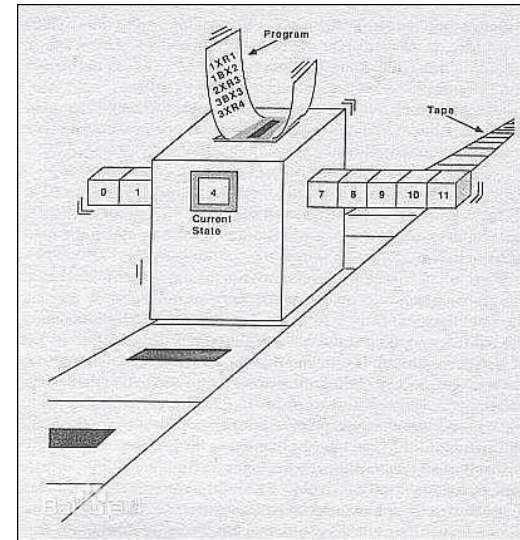
1.1 操作系统的概念



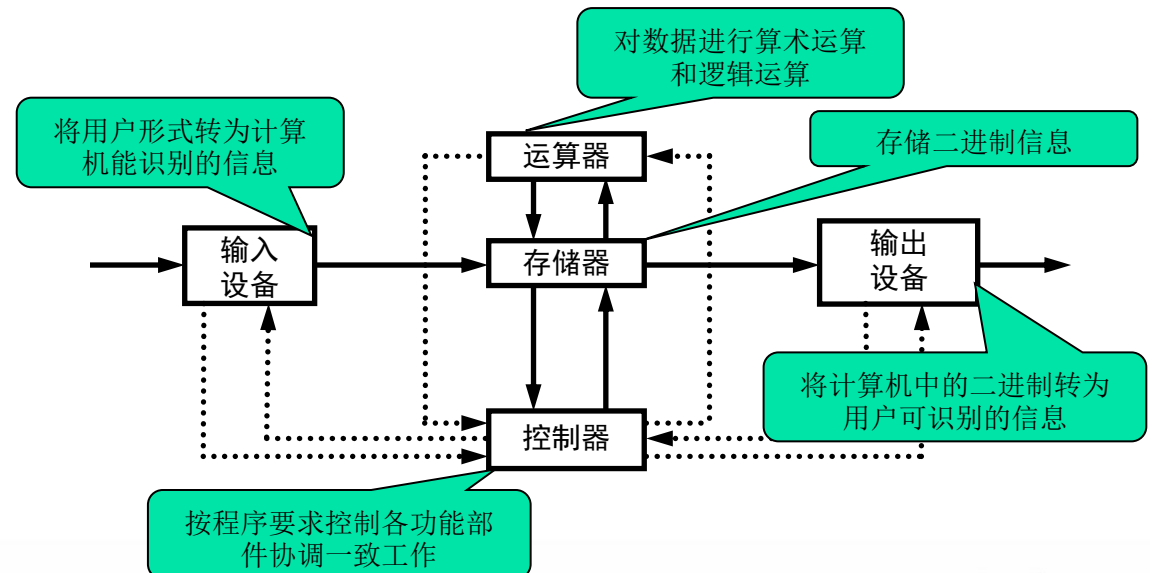
Alan Turing

图灵机模型

用机器来模拟人们用纸笔进行计算的过程（大脑里的计算机）



John von Neumann





计算机系统

- 计算机系统由硬件和软件两部分组成。
 - **硬件**：计算机系统中由电子、机械、电气、光学和磁学等元器件构成的各种部件和设备。CPU、存储器及设备都是硬件。
 - **软件**：完成一定任务的程序及其数据。包括系统软件及应用软件。系统软件有操作系统、编译程序、编辑程序、数据库管理系统等；应用软件是为各种应用目的而编制的程序。



计算机系统的层次关系

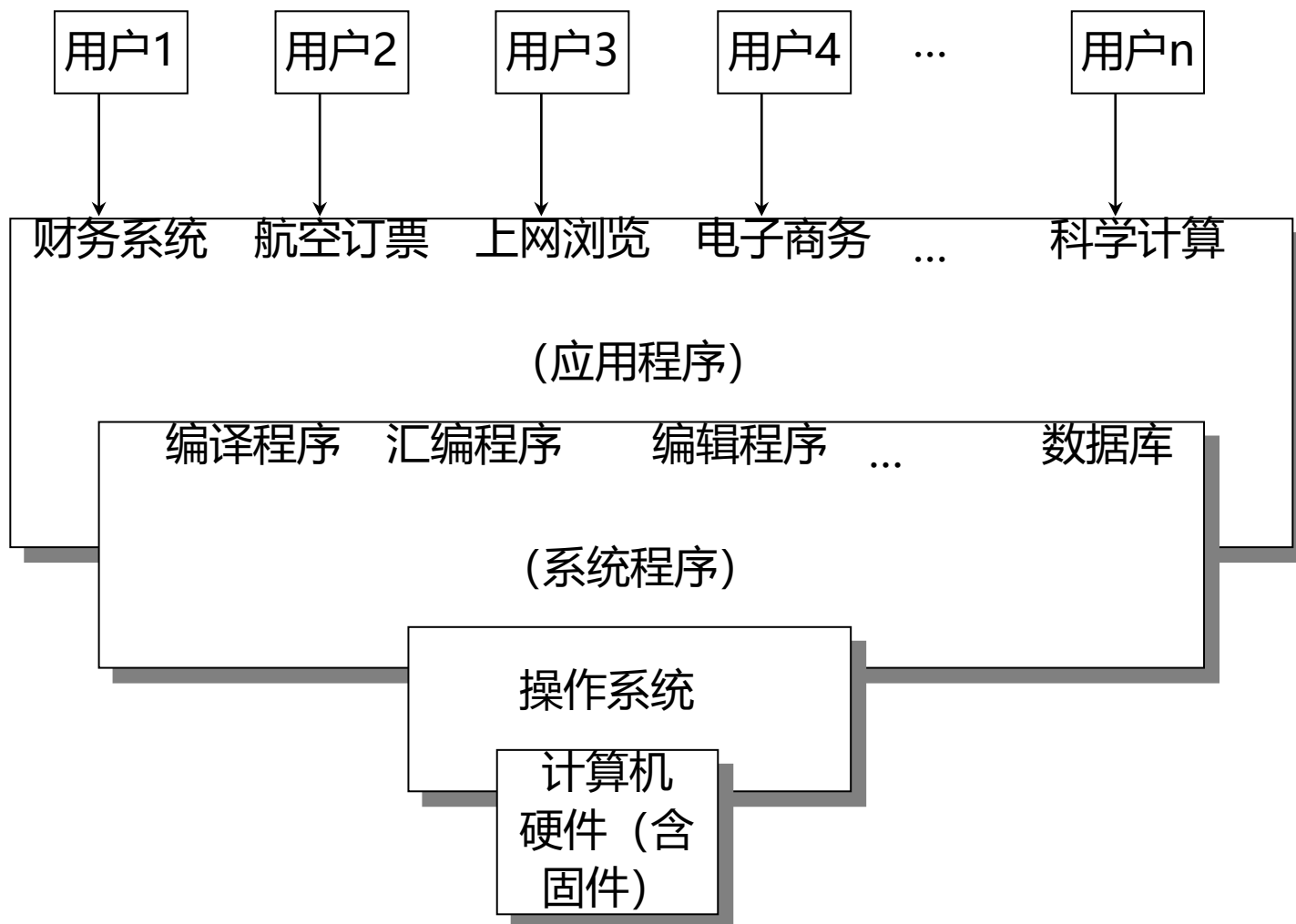
计算机硬件、软件，以及软件的各部分之间形成了一种**层次结构**的关系。

- **操作系统**是配置在计算机硬件上的第一层软件，是对硬件的首次扩充，形成了虚拟计算机/扩展机。
- 操作系统位于硬件与其它软件之间，是所有其他软件运行的基础。





计算机系统层次关系实例





为什么需要操作系统

◆ 从用户视角看：

- 操作系统是用户与计算机硬件之间的接口。为用户使用计算机提供服务。

◆ 从系统视角看：

- 操作系统是计算机系统资源的管理者。
 - 硬件管理、软件管理
- 对计算机的能力进行扩展，在不影响原有服务前提下，有效引入新的功能
- 实现：资源复用、资源虚化以及资源抽象

何为复用、虚化与抽象？



几个概念：

■ 资源复用：

- 让系统对宝贵资源能够共享地来使用
- 解决资源数量不足的问题
- 资源复用可分为：
 - 时间复用：多个用户或程序轮流使用某个资源，如CPU使用
 - 空间复用：多个用户或程序同时使用资源的一部分，不需排队，如内存使用

■ 资源虚拟化：

- 让一个物理设备变成多个相对独立的对应物，也称为虚拟性
- 为资源的复用提供了技术支持



几个概念：

■ 资源抽象：

- 化繁为简，把具体的资源管理，变成较为统一、简单的管理方式
- 通过创建新的管理对象来屏蔽底层资源的复杂特性和接口细节
- 对内封装细节、对外提供管理对象与访问接口
- 如：面向进程而不是CPU，面向文件而不是物理存储介质，面向窗口而不是屏幕

复用与虚拟化解决的是资源不足的问题

资源抽象解决的是资源复杂的问题

复用、虚化、抽象是结合在一起进行使用的



操作系统的定义?

- **没有公认且统一的精确定义**
- **操作系统是一个控制程序**
 - 一个系统软件
 - 控制程序执行过程，防止错误和计算机的不当使用
 - 执行用户任务，给用户提供各种服务
 - 方便用户使用计算机系统
- **操作系统是一个资源管理器**
 - 应用程序与硬件之间的中间层
 - 管理各种计算机软硬件资源
 - 提供访问计算机软硬件资源的高效手段
 - 解决资源访问冲突，确保资源公平使用



操作系统的定义

- **操作系统**是计算机系统最基础的系统软件，管理软硬件资源、控制程序执行，改善人机界面，合理组织计算机工作流程，为用户提供使用计算机提供良好运行环境。
- **操作系统**是一组控制和管理计算机硬件和软件资源、合理地各类作业进行调度，以及方便用户的程序的集合。
- ...



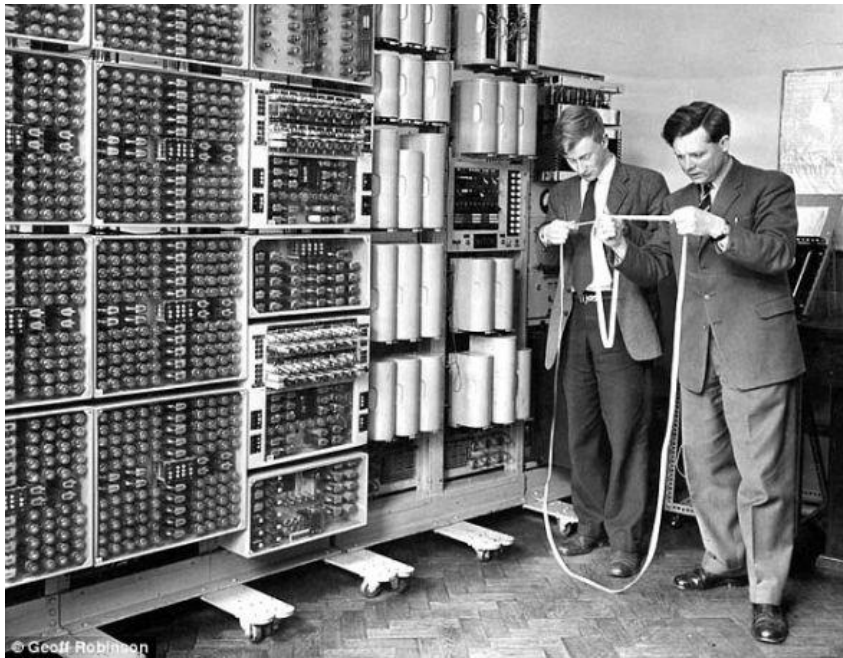
1.2 操作系统的形成与发展

- 操作系统的发展过程是一个从无到有，从简单到复杂的过程。
- 最初计算机上无操作系统，20世纪50年代出现了简单批处理系统，60年代出现了多道批处理系统，不久又出现了分时系统及实时系统...



1.2.1 手工操作阶段

- **人工操作方式：** 46~50年代中后期，计算机系统上没有配置操作系统，人们使用计算机采用手工操作方式。



人工操作不足：

- 人工慢速与CPU运算的高速之间的矛盾。人机矛盾
- CPU的快速与I/O设备慢速的矛盾。
- 用户独占计算资源，资源利用率低
- 手工操作容易出错



1.2.2 批处理系统

- 为解决人机矛盾，人们提出了从一个作业到下一个作业的**自动过渡方式**，从而出现了**批处理技术**。

如何实现自动？

- **监督程序 (Monitor)** 是一个常驻内存的程序，它管理作业的运行，负责装入和运行各种系统程序来完成作业的自动过渡。
- 监督程序是最早的操作系统雏形。



批处理系统

■ 工作方式

1. 用户将作业交给系统操作员
2. 系统操作员将许多用户的作业组成一批作业，输入到计算机系统中，在系统中形成一个自动转接的连续的作业流
3. 启动监督程序
4. 系统自动、依次执行每个作业
5. 由操作员将作业结果交给用户

■ 追求目标：

- 提高资源利用率，增加作业处理吞吐量



联机批处理和脱机批处理

■ 联机批处理

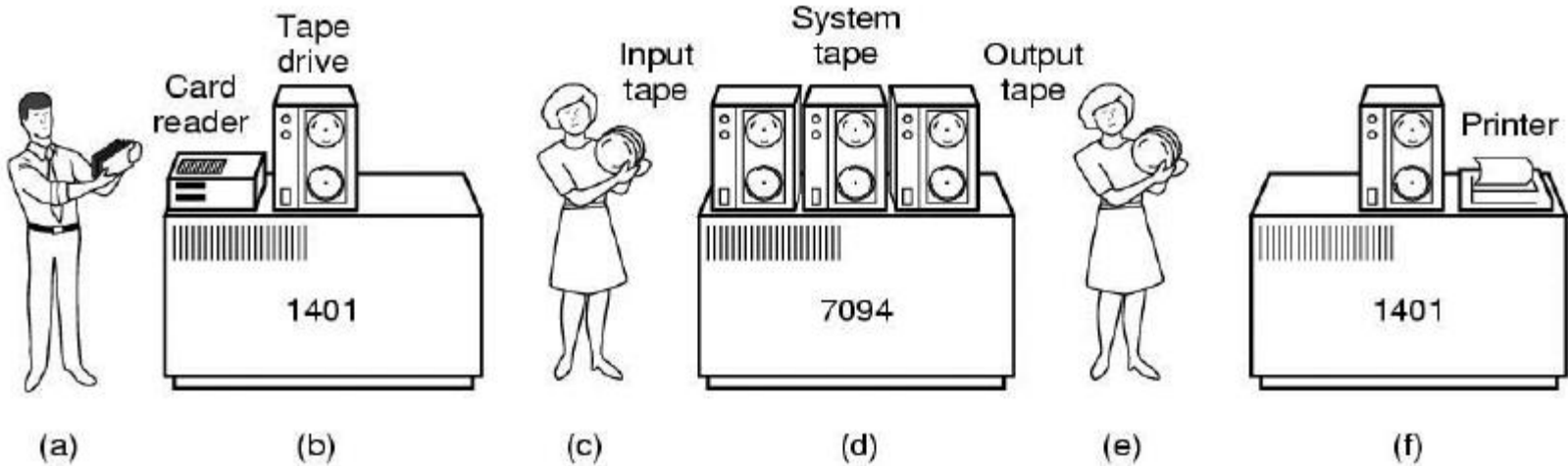
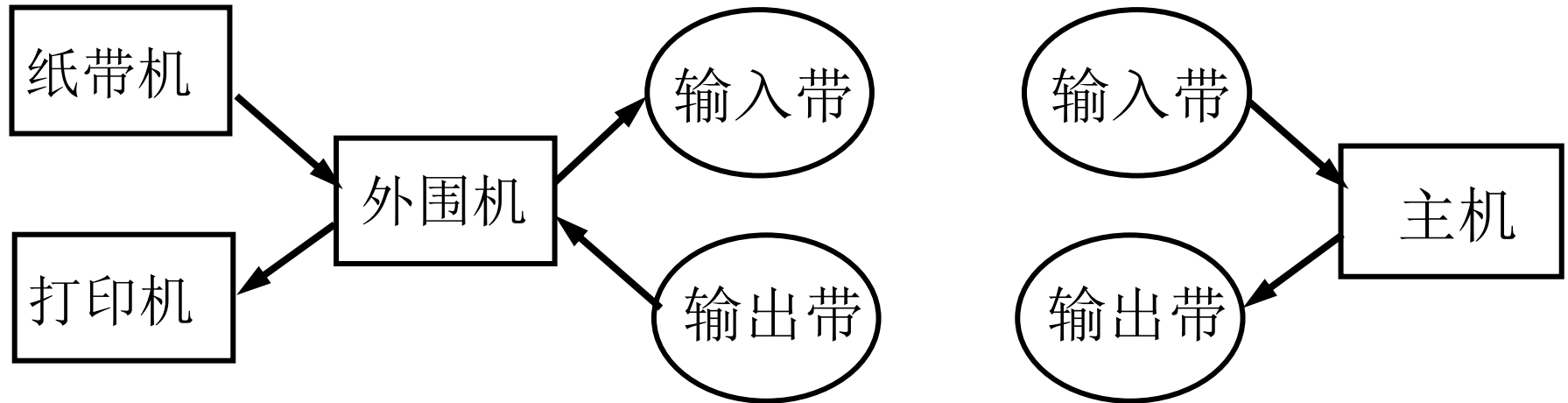
- 采用**联机输入/输出**，即输入/输出操作在监督程序控制下，**同步**进行
- 缺点：**速度慢**

■ 脱机批处理

- 采用**脱机输入输出**
 - 外围机：独立于主机，负责把作业任务写入磁带，以及从磁带输出到打印机。
 - 主机：负责从磁带上把作业调入内存，或者把结果写回磁带。
- 优点：
 - 实现了部分**异步**能力
 - **减少了CPU的空闲时间**
 - **提高I/O速度**



脱机输入/输出示意图





单道批处理系统及其特征

- 早期的批处理系统内存中始终只保持一道作业，这样的批处理系统也称为**单道批处理系统**。
- 单道批处理系统的特征：
 - **自动性**：自动依次运行，无人工干预
 - **顺序性**：完成顺序与进入顺序相同
 - **单道性**：内存只有一道作业



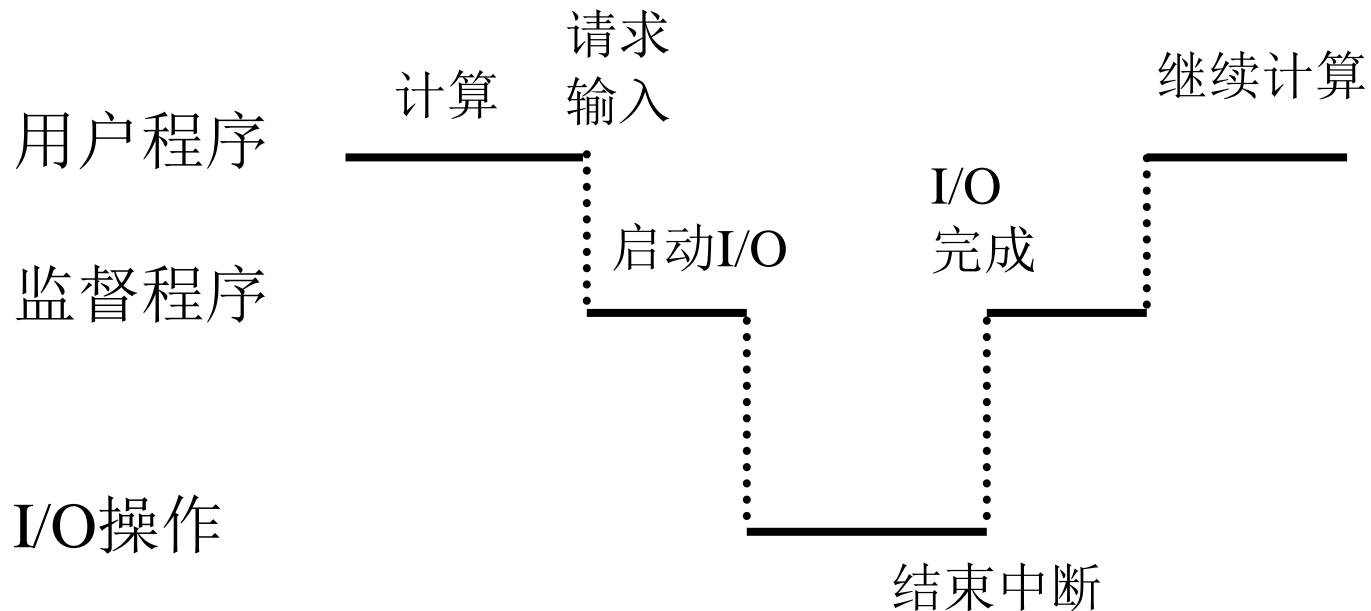
多道批处理系统及其特征

- 在批处理系统中引入多道程序设计技术后就形成了多道批处理系统。
- 多道批处理系统的特征：
 - **多道性**：计算机内存中同时存放几道相互独立的程序。
 - **无序性**：进入顺序与完成顺序无严格对应关系。
 - **调度性**：作业从提交到完成经历两级调度：作业调度、进程调度。



多道程序设计技术

- 单道批处理系统中内存仅一道程序，系统资源无法得到充分利用。下图是单道程序运行实例





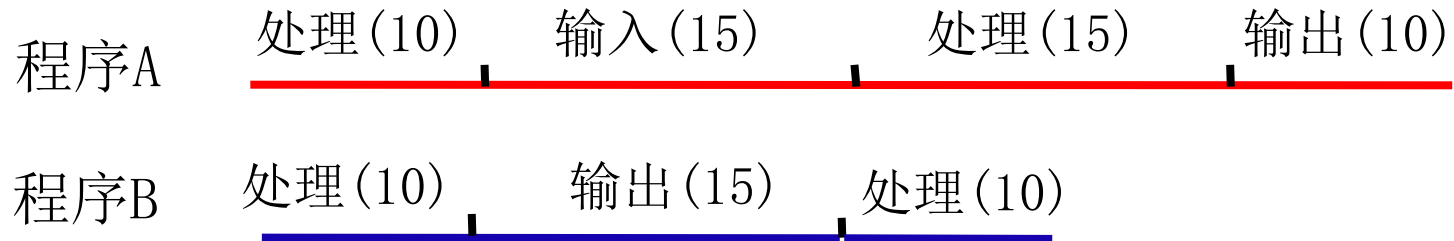
多道程序设计技术

- 多道程序的核心思想：
 - 将多个作业存放在主存中，这些程序在管理程序的控制下交替运行，共享处理机和系统中的其他资源。

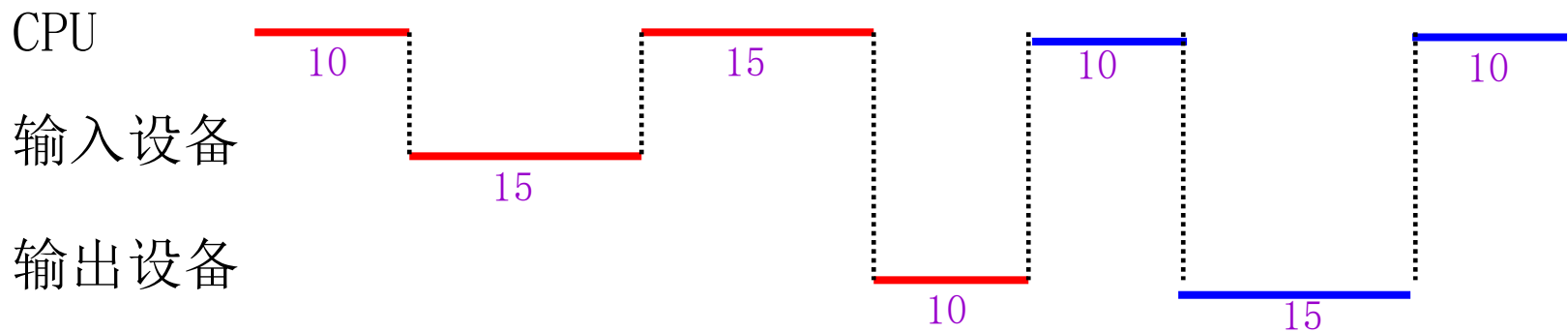


多道程序运行实例1

■ 有程序A和程序B



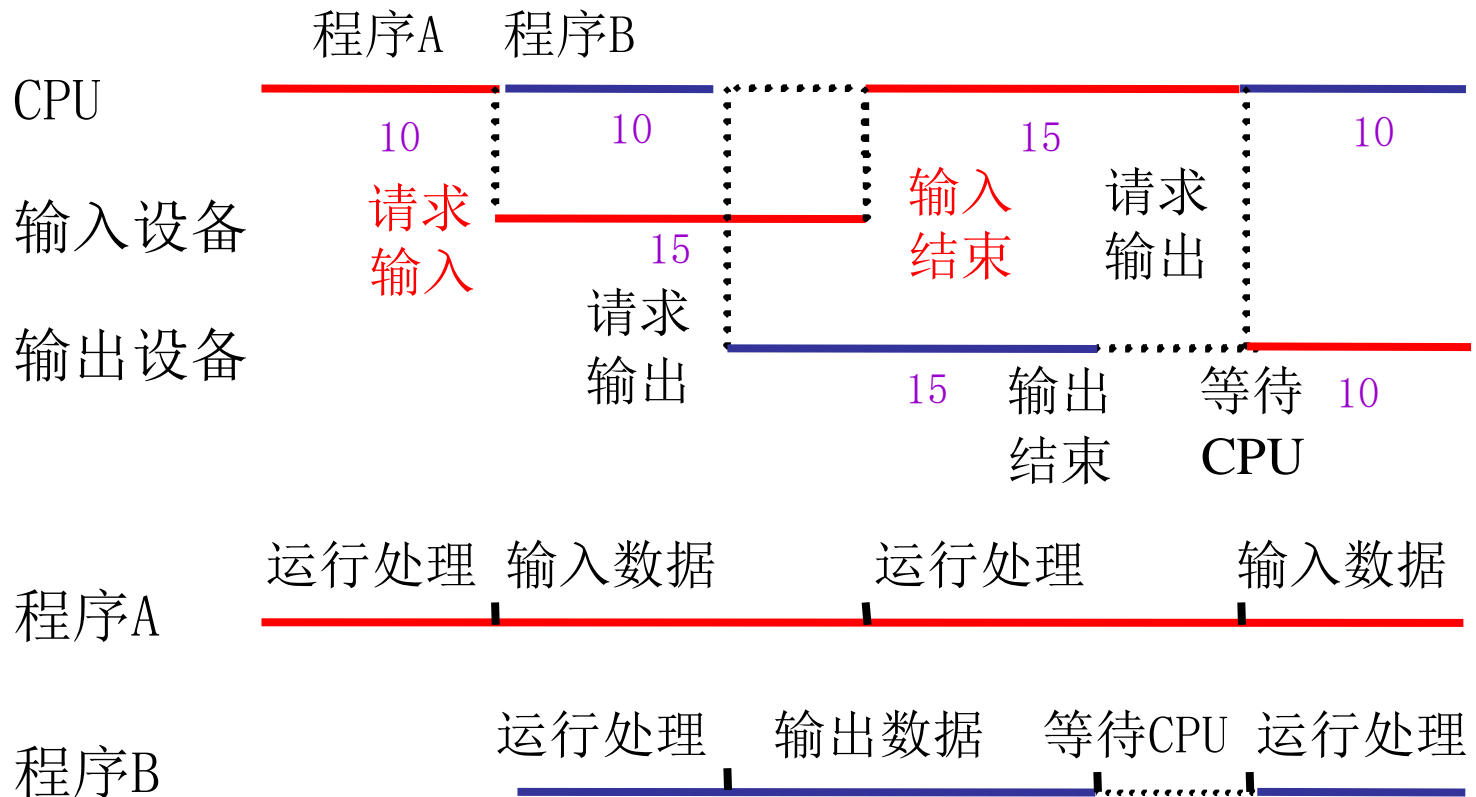
■ 单道程序运行方式:





多道程序运行实例1

■ 多道程序运行方式:





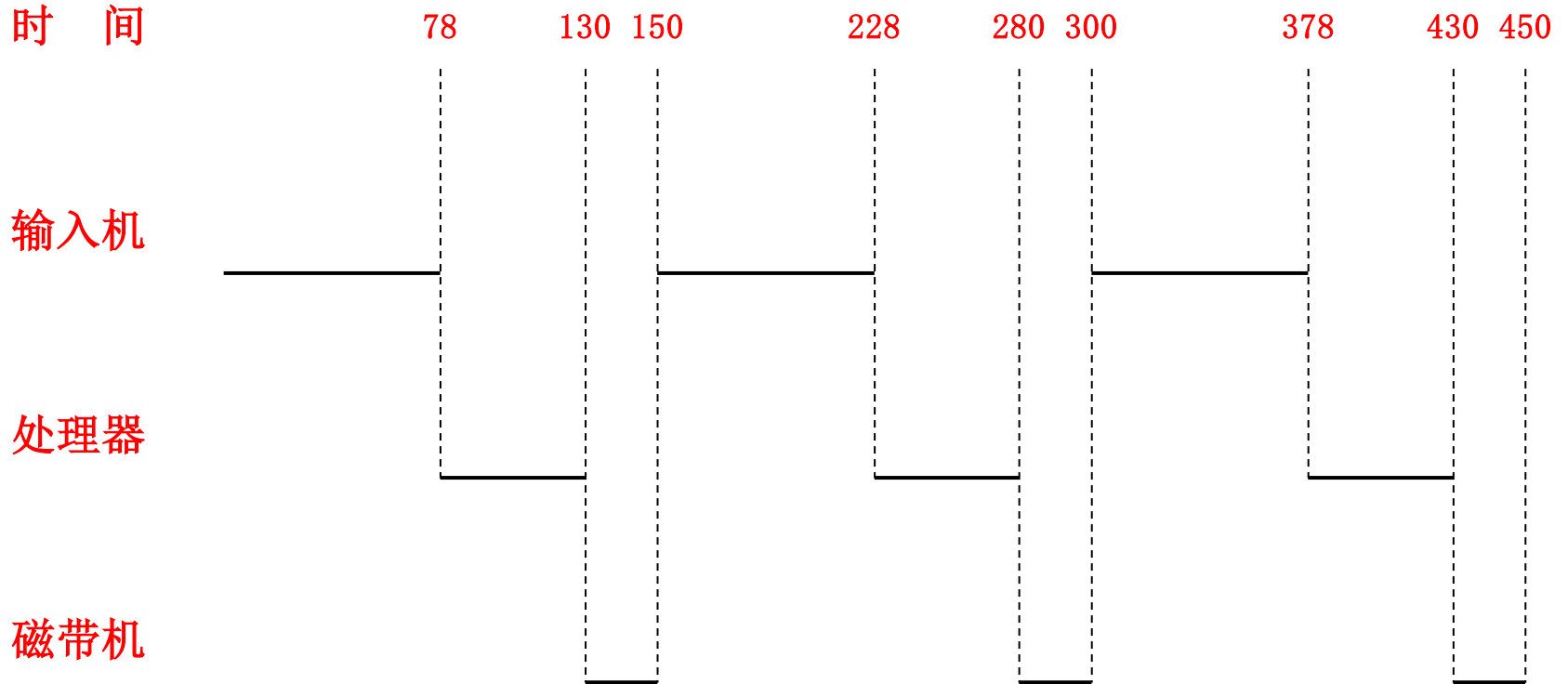
多道程序运行实例2

- 计算某个数据处理问题，要求从输入机（速度为6400字符/秒）输入500个字符，经CPU处理后（约52ms），将结果（假定为2000个字符）存到磁带上（磁带机速度为10万字符/秒），重复再读500个字符，如此往复进行，直至输入数据全部处理完毕
 - 输入时间：78ms
 - 输出时间：20ms
 - 处理时间：52ms



多道程序运行实例2

■ 单道程序运行



■ 处理器的利用率为

■ $52 / (78 + 52 + 20) \approx 35\%$



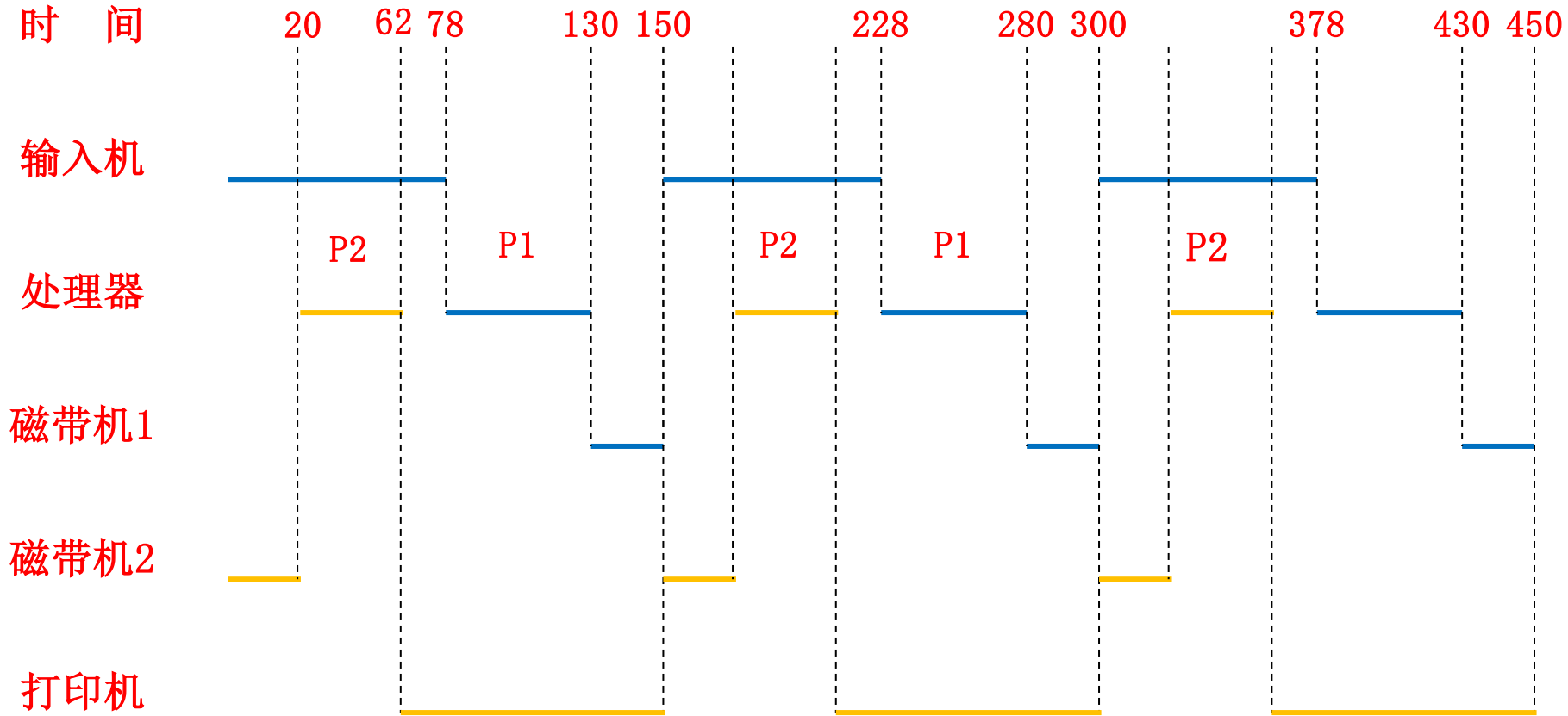
多道程序运行实例2

- 计算机还接受了另一算题：从另一台磁带机（磁带机速度为10万字符/秒）上输入2000个字符经42ms的处理后，从行式打印机（速度为1350行/分）上输出两行
 - 输入时间：20ms
 - 处理时间：42ms
 - 输出时间：88ms
- 如果是该任务单道运行，处理器利用率是：
 - $42/(20+42+88)=28\%$



多道程序运行实例2

■ 两道程序并发运行



■ 处理器的利用率

■ $(52+42) / (78+52+20) \approx 63\%$



多道程序运行特点

- **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**：
 - 同时进入系统的多道程序都处于运行过程中，即它们先后开始了各自的运行，但都未运行完毕。
- **微观上串行**：
 - 内存中的多道程序轮流占有 CPU，交替执行。



多道程序设计的优点与缺点

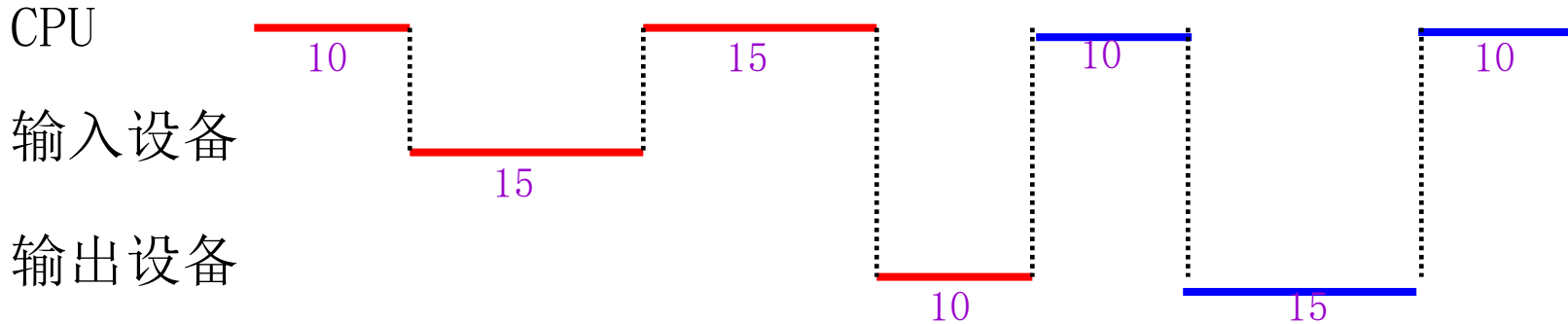
- 提高了CPU的利用率
- 提高了内存和I/O设备的利用率
- 改进了系统的吞吐量*
- 充分发挥了系统的并行性
- 其主要缺点是: 作业周转时间*延长

*吞吐量是单位时间完成的作业道数

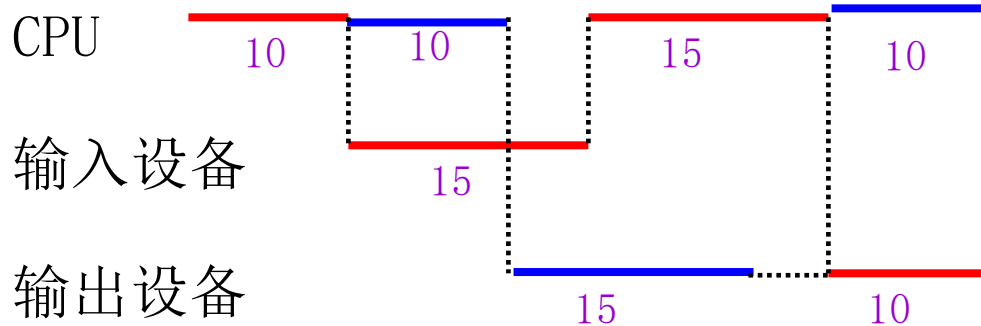
*作业的周转时间是指作业从进入系统开始, 直至其完成并退出系统为止所经历的时间。



例：单道程序VS多道程序设计技术



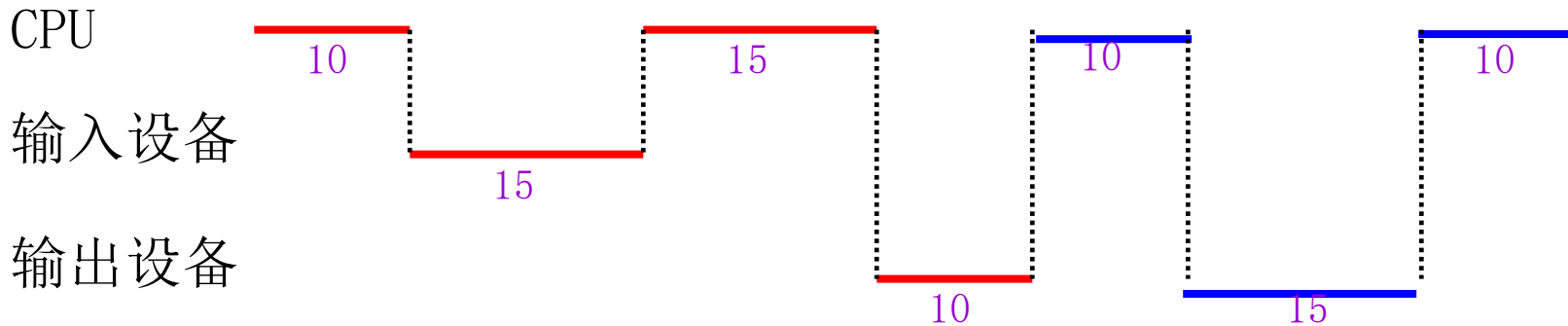
CPU利用率:? 平均周转时间:? 系统吞吐量:?



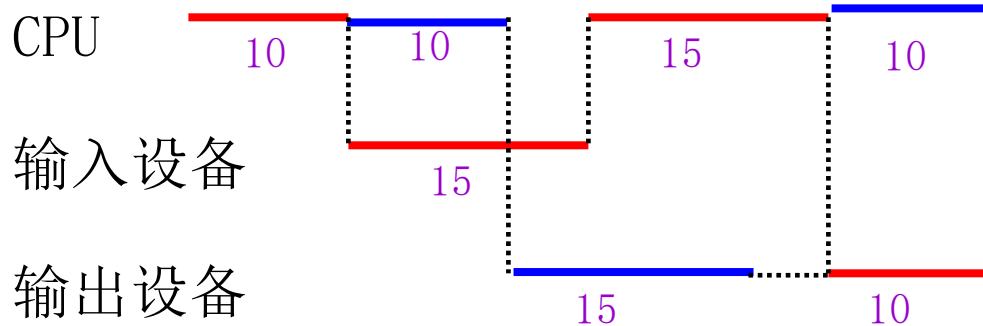
■CPU利用率:? 平均周转时间:? 系统吞吐量:?



练习：单道程序VS多道程序设计



CPU利用率: $(10+15+10+10)/85=53\%$ 平均周转时间: $(50+35)/2=42.5$ 系统吞吐量: $2/85=0.0235$



CPU利用率: $(10+15+10+10)/50=90\%$ 平均周转时间: $(50+40)/2=45$ 系统吞吐量: $2/50=0.04$



多道程序设计的道数问题

- 多道程序设计的道数多少**不是任意的**
 - 例如，两道程序都要用打印机，若系统只有一台，就算它们同时接受进入计算机内存运行，未必能充分提高效率。
- 内存储器的容量和用户的响应时间等因素也影响多道程序道数的多寡。



多道程序设计的道数问题

- 假如一道程序平均等待I/O操作的时间占其运行时间的比例为 p ，当内存中有 n 道同样的程序时，所有程序都等待I/O的概率是 p^n ，此时CPU是空闲的，那么：

$$\text{CPU利用率} = 1 - p^n$$

n 称多道程序的道数或度数，可见CPU的利用率是 n 的函数



一个例子

- 主机主存：1MB，操作系统内核占用：200KB，每道程序占用：200KB，假设程序等待I/O平均时间占80%，如果忽略操作系统开销，则cpu利用率情况如何？
 - 理想情况可容纳4道程序，CPU的利用率为： $1 - 0.8^4 = 59\%$
 - 如果再提高1MB内存，则利用率为？
 - 87%，相比前次提高了47%
 - 如果继续增加1MB内存，利用率为96%，比前次提高了10%



注意：多重处理系统

- 多重处理系统是指配置了多个物理CPU，能真正同时执行多道程序的系统。
 - 只有一个处理器的系统称为单处理器系统
- 要有效地使用多重处理系统，必须采用多道程序设计技术；反过来，多道程序设计不一定要求有多重处理系统支持。



多道批处理需要解决的问题

- 处理机管理：如何分配回收处理机
- 内存管理：内存分配回收，存储保护，程序浮动
- 设备管理：设备分配回收、缓冲管理
- 文件管理：文件存储、共享保护及检索
- 作业管理：作业合理搭配（如CPU型与I/O型）



1.2.3 分时系统

- 允许多个联机用户同时使用一台计算机系统进行计算的操作系统称分时操作系统
- 推动分时系统发展的主要动力是用户需求，即：
 - 人机交互：控制程序运行，提高用户参与度
 - 共享主机：机器昂贵，多人使用较经济
 - 方便用户上机： 可以通过终端提交作业



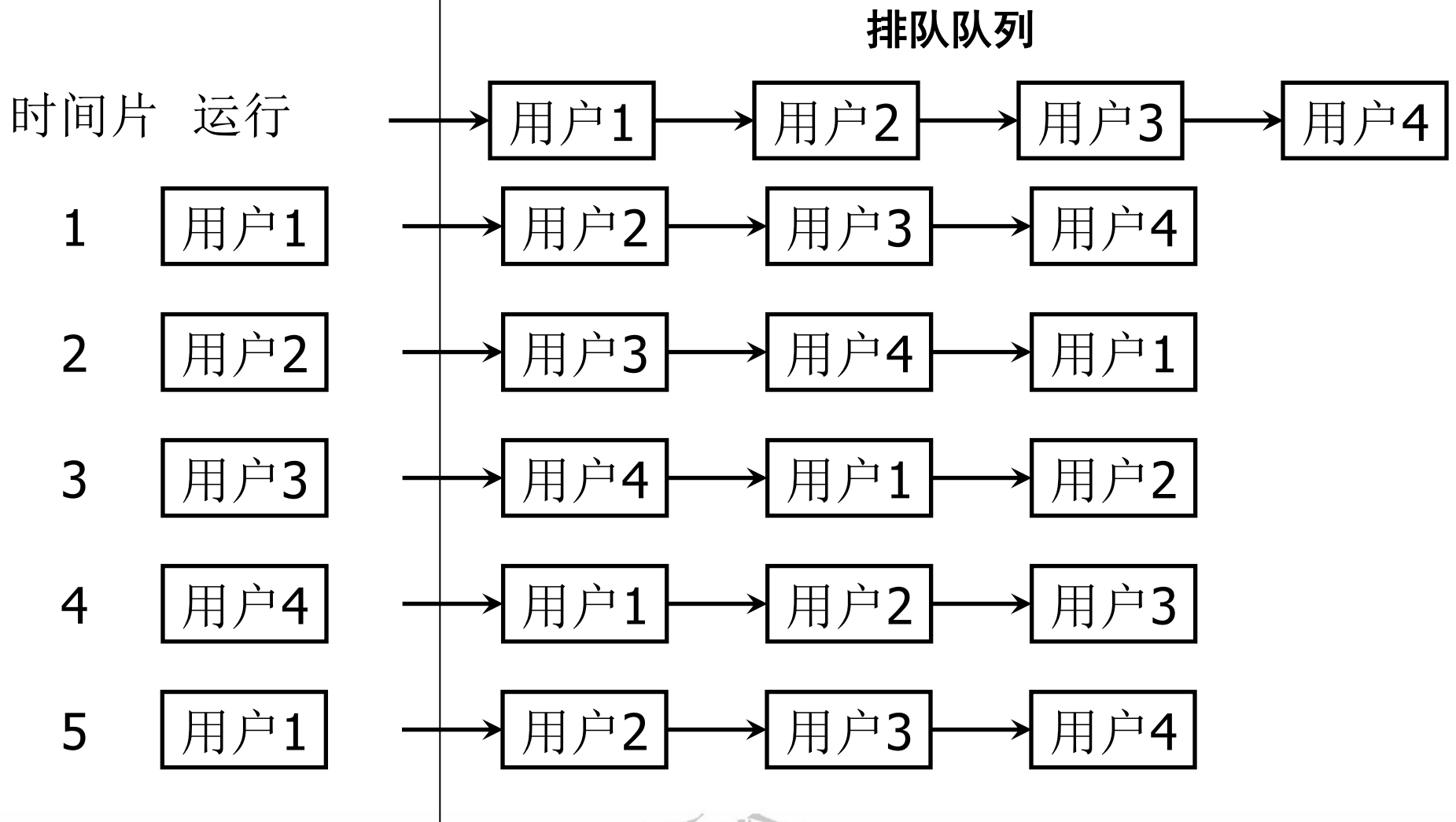
分时技术

- 所谓**分时技术**就是：
 - 把处理机的运行时间分成很短的时间片，按时间片轮流把处理机分配给各联机作业使用。
 - 若某个作业在分配给它的时间片内不能完成其计算，则该作业暂停运行，把处理机让给另一个作业使用，等待下一轮时再继续其运行。
 - 一个例子：为四个用户服务



分时技术示意图

- 假设系统中有4个联机用户，按如下方式排列：





分时操作系统

- 在操作系统中采用分时技术就形成了**分时操作系统**。
 - 关键问题
 - 及时接收：设置多路卡
 - 及时处理：时间片轮转
- 分时系统的历史
 - 1959由MIT提出，并于1962年开发出第一个分时系统CTSS (Compatible Time Sharing System)，运行于IBM 7094，支持32个用户
 - 1965年，IBM 在IBM360上开发了TSS/360，太大太慢、没用户。
 - 1965年，美国 DoD 支持下，MIT, Bell, GE 开发了MULTICS (MULTiplexed Information and Computing Service),运行于GE636\GE645



分时操作系统的特征

- **多路性**：也称**同时性**，一台计算机与若干台终端相连接，终端上的这些用户可以同时使用计算机。
- **交互性**：用户通过终端采用人机会话的方式直接控制程序运行，同程序进行交互。
- **独立性**：用户彼此之间都感觉不到别人也在使用这台计算机，好像只有自己独占计算机一样。
- **及时性**：用户请求能在很短时间内获得响应。



1.2.4 实时系统

■ 实时系统

- 指系统能及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，并控制实时任务协调一致地运行。

■ 主要分类

- **实时控制系统**：指以计算机为中心的生产过程控制系统。（生产过程控制、飞机导弹控制）
- **实时信息处理系统**：要求对信息进行实时处理的系统。（银行交易系统、网络查询系统）



实时系统的特征

- **及时性**：响应时间由控制对象决定。
- **可靠性**：高可靠性。

思考&讨论：

- 对实时系统的攻击可能有哪些？



通用操作系统

- 操作系统有三种基本类型：
 - 批处理操作系统
 - 分时操作系统
 - 实时操作系统
- 如果一个操作系统兼有批处理、分时和实时操作系统三者或其中两者的功能，则称该操作系统为通用操作系统。



1.2.5 操作系统的进一步发展

- 嵌入式操作系统
- 个人计算机操作系统
- 网络操作系统
- 分布式操作系统
-



嵌入式操作系统

- 对整个智能芯片以及它所控制的各种部件模块等资源进行统一调度、指挥和控制的系统软件称为**嵌入式操作系统**。
- 嵌入式系统几乎包括了生活中的所有电器设备，如掌上PDA、手机、微波炉、数码相机、自动售货机、工业自动化仪表与医疗仪器等。
- 嵌入式系统进一步延伸，目前有移动操作系统、手机操作系统、物联网系统、工业控制系统等。



个人计算机操作系统

- **个人计算机操作系统**主要供个人使用，它功能强，价格便宜，能满足一般人工作、学习、游戏等方面的需求。
- 个人计算机操作系统的主要特点是计算机在某一段时间内为单个用户服务，采用图形界面人机交互的工作方式，界面友好，使用方便。



网络操作系统

- **网络操作系统**是基于计算机网络的，是在各种计算机操作系统上按网络体系结构协议标准开发的软件
- 包括网络管理、通信、资源共享、系统安全和各种网络应用服务，其目标是互相通信及资源共享。



分布式操作系统

- 分布式系统是指多个分散的处理单元经互连网络连接而形成的系统，其中每个处理单元既具有高度自治性又相互协同，能在系统范围内实现资源管理、任务动态分配，并能并行地运行分布式程序。
- 配置在分布式系统上的操作系统称为**分布式操作系统**。
- 分布式系统继续发展，目前还有网格系统、云计算系统



操作系统发展的推动力

- 操作系统发展是随着计算机**硬件**技术更新、**应用**需求的发展、软件新技术的出现而发展的
- 目标：
 - 充分利用硬件
 - 提供更好的服务



1.3 操作系统的基本特征

- 操作系统有四个基本特征：
 - 并发性
 - 共享性
 - 虚拟性
 - 异步性（或不确定性）



1.并发性

- 并行（Parallelism）：是指两个或多个事件在同一时刻发生。
- 并发（Concurrency）：是指两个或多个事件在同一时间间隔内发生。
- 并行的事件或活动一定是并发的，但反之并发的的事件或活动未必是并行的。
- 并行性是并发性的特例，而并发性是并行性的扩展。



1.并发性

- 在多道程序环境下，并发是指宏观上一段时间内有多道程序在同时运行，但在单处理机系统中，微观上这些程序是交替执行的。



1.并发性

- 通常的程序是静态实体(Passive Entity), 并不能并发执行, 为了使得程序能够并发执行, 系统必须为每个程序建立运行实体, 即进程。
- 进程: 是在系统中能独立运行并作为资源分配基本单位的活动实体(Active Entity)。



并发性使系统变得复杂化

- 如何从一个活动切换到另一个活动？
- 怎样将各个活动隔离开来，使之互不干扰，免遭对方破坏？
- 怎样让多个活动协作完成任务？
- 怎样协调多个活动对资源的竞争？
- 如何保证每个活动的资源不被其它进程侵犯？
- 多个活动共享文件数据时，如何保证数据的一致性？



并发性的实现

- 采用并发技术的系统称为多任务系统 (Multitasking),
- 并发的实质是一个物理CPU(也可以多个物理CPU) 在若干道程序之间**多路复用**, **并发性**是对有限物理资源强制行使多用户共享以提高效率。
- **实现并发技术的关键之一**是如何对系统内的多个活动(进程)进行**切换**的技术



2.共享

- **共享**是指系统中的资源可供多个并发执行的进程共同使用。
- 有两种资源共享方式：
 - 互斥共享：一段时间只允许一个进程访问
 - 同时访问：一段时间允许多个进程访问
- 与共享性有关的问题:资源分配、信息保护、存取控制等，必须要妥善解决好这些问题



操作系统的两个最基本的特征

- 并发和共享是操作系统的**两个最基本特征**，二者之间互为存在条件。
 - 一方面，资源的共享是以程序的并发执行为条件的，若系统不允许程序的并发执行，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，也必将影响到程序的并发执行，甚至根本无法并发执行。



3.虚拟性

- **虚拟**是指把一个物理上的实体变为若干个逻辑上的对应物；或把物理上的多个实体变成逻辑上的一个对应物的技术
- 显然，前者（物理实体）是实际存在的，而后者（逻辑对应物）是虚构假想的



虚拟性的例子

1. 通过多道程序和分时使用CPU技术，物理上的一个CPU变成逻辑上的多个CPU
2. 通过窗口技术可把物理上的一个屏幕变成逻辑上的多个虚拟屏幕
3. IBM的VM技术把物理上的一台计算机变成逻辑上的多台计算机
4. 虚拟存储器则是把物理上的多个存储器(主存和辅存)变成逻辑上的一个(虚存)的例子。



4.异步性

- 异步性（或称为不确定性）表现为多个作业的执行过程是走走停停的，其执行顺序和每个作业的执行时间是不确定的。



操作系统中的异步性处处可见

1. 进程何时执行？何时暂停？怎样的速度向前推进？都是异步(随机)的。
2. 作业到达系统的类型和时间是随机的；
3. 操作员发出命令或按按钮的时刻是随机的；
4. 程序运行发生错误或异常的时刻是随机的；
5. 各种各样硬件和软件中断事件发生的时刻是随机的



异步性带来的问题

- 异步性给系统带来潜在危险，有可能导致与时间有关的错误。
- 操作系统的一个重要任务是必须确保捕捉任何一种随机事件，正确处理可能发生的随机事件，正确处理任何一种产生的事件序列，否则将会导致严重后果



1.4 操作系统的作用和功能

■ 1.4.1 操作系统的作用

1. OS作为用户与计算机硬件之间的接口
2. OS作为计算机系统的资源管理者
3. OS作为虚拟计算机（扩充机器）



1、OS是用户与计算机之间的接口

- 用户需要计算机能：
 - 易用性：对硬件实施改造，使之更易使用；
 - 可扩展性：提供系统调用，可供二次开发，扩展机器功能；
 - 提供运行环境：对工作流程进行组织，提供运行环境。
- 这些功能由硬件直接实现**很困难！**



1、OS是用户与计算机之间的接口

- 操作系统**改造和扩充**过的计算机不但功能更强，使用也更为方便，用户可直接**调用系统**提供的各种功能，而无需了解软硬件本身的细节，为用户提供一个与底层**细节无关**可管理的**运行环境**，对于用户来讲OS便成为他与计算机硬件之间的一个接口。



2、OS是计算机系统资源的管理者

- 操作系统管理的计算机系统软硬件资源包括：
 - 处理机
 - 存储器
 - 设备
 - 文件（程序和数据）
 - 也可把操作系统定义为：**是能使诸用户有效、方便地共享一套计算机系统资源的一种系统软件。**



2、OS是计算机系统资源的管理者

■ 资源管理

- 基本要求：监控与有序管理资源。
- 抽象资源：找出各种资源共性和个性，施加统一化监控
- 统一方法：使用最简单方法进行管理与共享资源



3、OS为用户提供虚拟计算机

- **如何把硬件复杂性与用户隔离开来?**
- **方法**
 - 在计算机裸机上加上一层层软件来组成整个计算机系统。
 - 每当在计算机上覆盖一层软件，**提供了一种抽象**，系统的**功能**便增加一点，**使用**就更加方便一点，用户可用的**运行环境**就更加好一点。



3、OS为用户提供虚拟计算机

- 操作系统是紧靠硬件的第一层软件，计算机上覆盖操作系统后，可扩展基本功能，为用户提供一台功能显著增强，使用更加方便，安全可靠性好，效率明显提高的机器，称为**虚拟计算机，或操作系统虚拟机 (Virtual Machine)**。



1.4.2 操作系统的功能

- 操作系统的主要功能包括：
 - 处理机管理
 - 存储器管理
 - 设备管理
 - 文件管理



1. 处理机管理

- 处理机管理的主要任务是对处理机的分配和运行实施有效的管理。
 - **进程控制**：负责进程的创建、撤消及状态转换。
 - **进程同步**：对并发执行的进程进行协调。有同步与互斥
 - **进程通信**：负责完成进程间的信息交换。
 - **调度**：分为作业调度和进程调度。
 - **作业调度**：从后备作业队列中按照一定的原则，选择若干作业进入内存。
 - **进程调度**：决定哪个进程获得处理机。



2.存储器管理

- 存储器管理的主要任务是方便用户使用存储器，提高存储器利用率，从逻辑上扩充内存。
- 存储器管理包括
 - 内存分配
 - 内存保护
 - 内存扩充
 - 地址映射



3.设备管理

- 设备管理功能包括：
 - **设备分配**：根据用户的I/O请求，为之分配所需的设备，设备使用完成后还应回收。
 - **缓冲管理**：对各类设备缓冲区进行有效管理。
 - **设备驱动**：主要完成设备启动、I/O操作及中断处理。
 - **设备独立性**：如何确保应用程序独立于物理设备；**如何虚拟化设备**。



4. 文件管理功能

- 文件管理的主要任务是对文件进行管理，方便用户使用并保证文件安全性。文件管理功能包括：
 - **文件存储空间的管理**：对文件存储空间进行管理，包括存储空间的分配与回收等功能。
 - **目录管理**：管理文件的数据结构，提供按名存取的功能。
 - **文件操作管理**：从外存读入数据或将数据写入外存。
 - **文件保护**：防止未授权用户存取文件；防止授权用户以不正确方式存取文件。



1.5 操作系统提供的服务和接口

1.5.1 OS提供的公共和基本服务

1. **程序执行**：系统能够把程序装入内存并运行。
2. **输入/输出操作**：系统统一管理设备，为用户程序的运行提供I/O服务。
3. **信息保存（文件系统管理）**：系统为用户提供文件读写等服务功能。
4. **通信服务**：提供进程间通信服务。
5. **错误检测及报告**：能对用户程序运行过程中出现的错误进行检测并及时报告给操作员或用户。
6. **资源分配**：为进程的运行分配资源，如CPU等。
7. **统计**：统计用户使用资源的类型和数量。
8. **保护**：对计算机中存储的信息进行保护，确保对资源的访问是受控的。

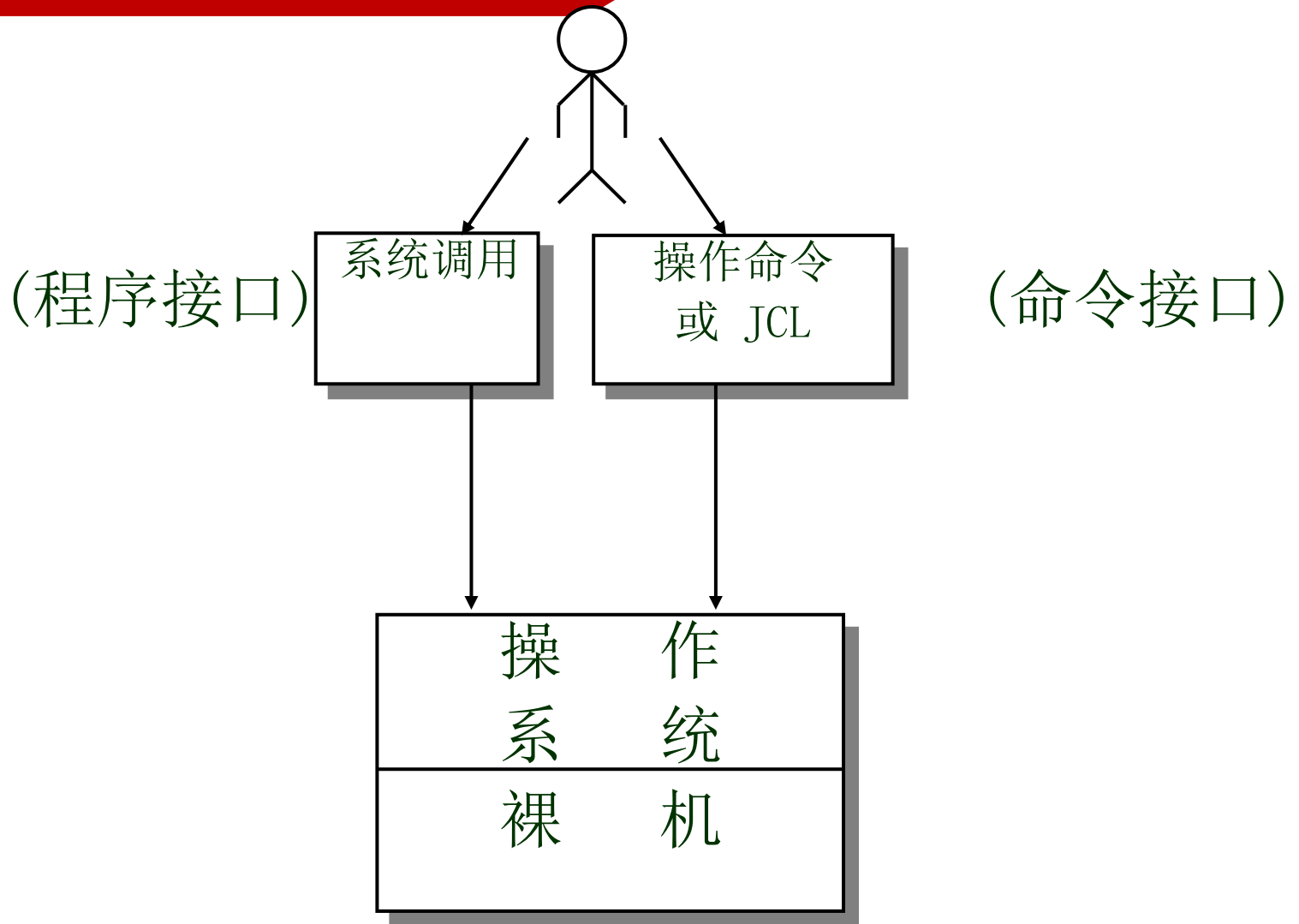


1.5.2 操作系统提供的接口

- 操作系统向用户提供了各种使用其服务功能的手段，即提供了**操作系统接口**。
- 操作系统向用户提供了三类接口：
 - 命令接口（或称操作接口）
 - 程序接口
 - 图形接口（图形式命令接口）



操作系统提供的接口





1. 命令接口

- 用命令接口进行作业控制的主要方式：
 - **脱机控制方式**：用户将对作业的控制要求以**作业控制说明书**的方式提交给系统，由系统按照作业说明书的规定控制作业的执行。
 - **联机控制方式**：指用户利用系统提供的一组键盘命令或其他操作命令和**系统会话**，交互式地控制程序的执行。



脱机命令接口

- **脱机命令接口**由一组作业控制语言(JCL, Job Control Language)组成。
- 脱机用户不能直接干预作业的运行, 他们应事先用作业控制命令写一份作业操作说明书, 连同作业一起提交给系统。
- 当系统调度到该作业时, 由系统中的命令解释程序对作业说明书上的命令或作业控制语句逐条解释执行。



联机命令接口

- **联机命令接口**提供一组命令供用户请求计算机系统服务。
- 联机命令也需要命令解释程序来进行处理，此处起作用为获取并执行用户指定的下一条命令。
 - 如Unix 中的Bourne Shell等
- 键盘命令可分为：
 - **内部命令**：命令功能简单、程序短小、使用频繁。它们在系统初始启动时被引导至内存并常驻内存。
 - **外部命令**：命令功能较复杂、程序较长、独立作为一个文件驻留在磁盘上，当需要它们时，再从磁盘上调入内存运行。



2. 图形用户接口GUI

- GUI是以基于鼠标的窗口和菜单系统为接口。
- 引入桌面概念，使用图像/图标代表程序、文件、目录和系统功能。
- 通过移动鼠标，把指针定位到桌面的图标上，实现对象命令与被操作对象的关联。
- 通过点击鼠标按钮，实现对屏幕对象的控制和操纵。
- 目前图形用户接口是最为常见的人机接口形式，可以认为图形接口是**命令接口的图形化**。
 - 1973，Xerox PARC中出现
 - Macintosh中实现普及→MacOS Mojave
 - MS-DOS → Windows 1.0
 - KDE & GNU GNOME



3. 程序接口

- 程序接口由一组**系统调用**命令组成。
- 用户通过在程序中使用这些系统调用命令来请求操作系统提供的服务。



系统调用

- **系统调用**：由若干条指令构成的过程，用以实现特定的操作系统服务功能。
- 系统调用命令有时也称为**广义指令**。它是由操作系统提供的一个或多个子程序模块实现的。



系统调用分类

- 按功能可以将系统调用分为：
 - 设备管理：完成设备的请求或释放、以及设备启动等功能。
 - 文件管理：完成文件的读、写、创建及删除等功能。
 - 进程控制：完成进程的创建、撤消、阻塞及唤醒等功能。
 - 进程通信：完成进程之间的消息传递或信号传递等功能。
 - 内存管理：完成内存的分配、回收以及获取作业占用内存区大小及始址等功能。



系统调用的实现要点

- **编写系统调用处理程序**；
- 设计一张系统调用**入口地址表**，每个入口地址都指向一个系统调用的处理程序，有的系统还包含系统调用自带参数的个数；
- 陷入处理机制需**开辟现场保护区**，以保存发生系统调用时的处理器现场。

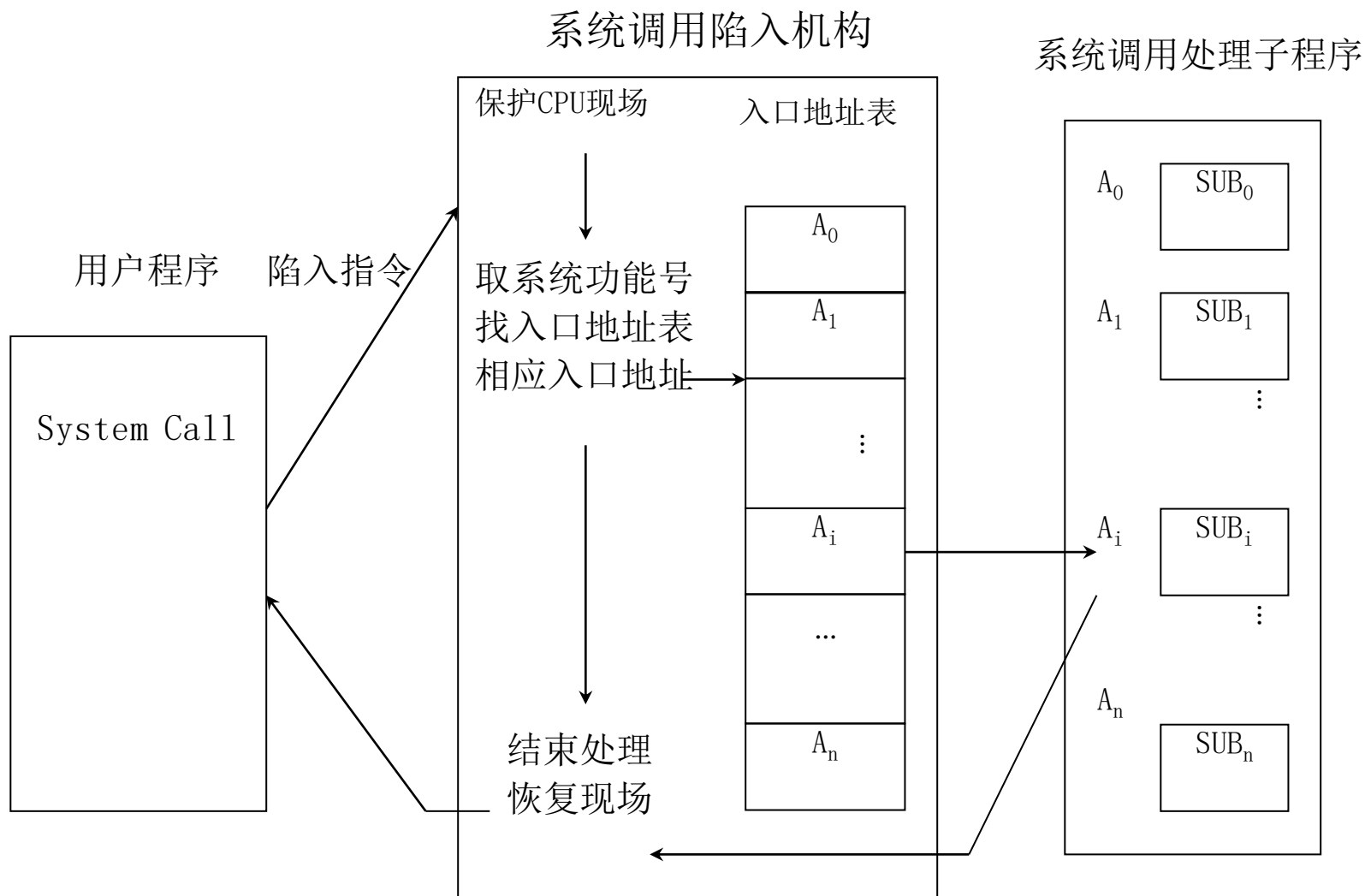


系统调用的调用过程

- 为执行系统调用命令作准备。主要工作是保留现场，并把系统调用命令的参数放入指定的存储单元。
- 执行系统调用。根据系统调用命令的编号找到相应子程序的入口地址，然后转去执行。
- 系统调用命令执行完后的处理。主要工作是恢复现场，并把系统调用的返回参数送入指定存储单元。



系统调用的处理过程





系统调用与过程调用的区别

- 运行状态不同：系统调用在**核心态**下运行，子程序在**用户态**下运行。
- 进入方式不同：系统调用通过中断机构进入以实现运行状态的改变，子程序直接调用不涉及运行状态改变。



1.6 操作系统的运行环境和内核结构

1.6.1 操作系统的运行环境

- 计算机硬件所提供的支持，构成了现代操作系统的运行环境。包括：
 - 处理机、存储器、设备、时钟、中断等
- 操作系统根据运行环境可分为
 - 系统初启程序：由时钟中断、外设中断所驱动
 - 服务程序：面向目态程序(Problem Mode，或用户态)，中断驱动
 - 系统内部模块：管态运行(Supervisor Mode，或核心态)，中断驱动



1.6.2 操作系统的内核结构

- 内核是操作系统的构件，或称基本单位。
- 内核作用：
 - 为进程提供管理：进程难于自己管理自己，同时也难于用硬件实现
 - 需要一个软件对硬件处理器及相关资源进行首次改造，为进程执行提供良好的运行环境。



内核的基本功能及属性

■ 基本功能:

1. 中断处理: 截获中断, 转向中断处理例程
2. 短程调度: CPU调度, 保存与恢复现场
3. 原语管理: 原语是不可中断过程

■ 基本属性:

1. 内核是由中断驱动的
2. 内核的执行是连续的
3. 内核在屏蔽中断状态下执行
4. 内核可以使用特权指令



操作系统的内核结构

- 操作系统是一个大型系统软件，其内核结构主要有四种：
 - 模块结构
 - 层次结构
 - 微内核结构
 - 虚拟机结构



1、模块结构

- **模块结构，又称为整体式结构**，将操作系统内核按照功能划分为一个个独立的模块，模块之间相对独立，只能通过事先规定好的接口方式来调用。
- 每个模块实现一个完整独立的功能，所有模块之间相互调用，共同构成一个完整的系统内核。



模块结构特点

- 效率高;
- 但全局函数使用多造成访问控制困难;
- 结构不清晰, 可理解性、可维护性及可移植性差。



2、层次结构

- **层次结构**是将操作系统内核按照一定的规则划分为一系列相互依赖的层次，每个层次也可以分解为一系列更小的模块，
- 每个模块完成一个特定的功能，只能与相邻层次发生直接联系，所有这些层次的集合就实现了整个系统。



层次结构特点

- 层次结构是一种特殊的模块结构。
- 其特点为：
 - 给模块赋予了层次顺序，使调用关系变得有序；
 - 在上下两层不变的基础上可以换掉某层，便于移植和扩充。
 - 但以牺牲一定的灵活性与通信开销为代价。
- 全序层次结构：层间单向依赖，层内模块独立
- 半序（偏序）层次结构：层间单向依赖，层内部分依赖



层次结构操作系统的分层原则

1. 把与机器硬件有关的程序模块放在最底层
2. 反映系统外特性的软件放在最外层
3. 按照实现操作系统命令时模块间的调用次序或按进程间单向发送信息的顺序来分层
4. 为进程的正常运行创造环境和提供条件的内核程序应该尽可能放在底层



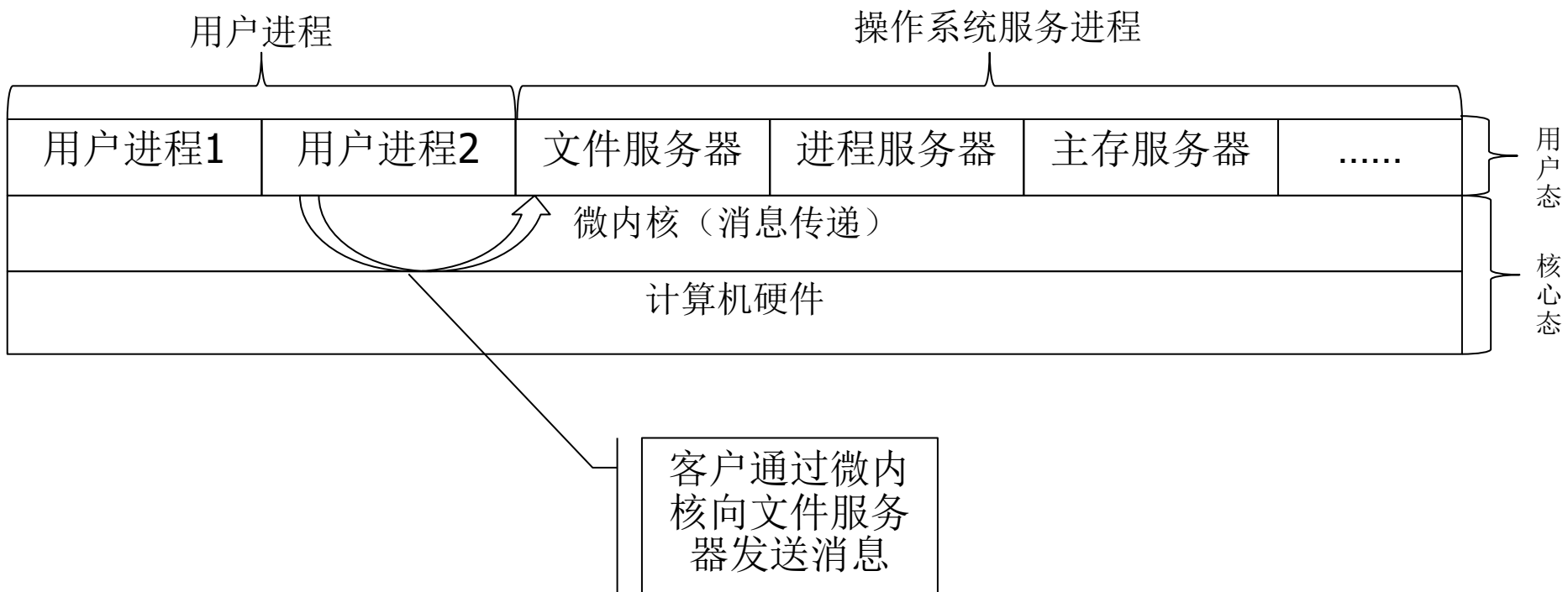
一个层次结构操作系统例子

- E. W. Dijkstra, 1986年提出的THE系统, 第一次提出层次结构设计方法, 运行于Electrologica X8:
 - 第0层完成中断处理、定时器管理和处理器调度
 - 第1层内存和磁鼓管理, 为进程分配内存空间, 并自动实现内存和磁鼓对换区的数据交换
 - 第2层处理进程与操作员间的通信, 为每个进程生成虚操作员控制台
 - 第3层I/O管理, 管理信息缓冲区
 - 第4层用户 (进程) 层
 - 第5层系统操作员 (进程) 层



3、微内核结构

- 核心思想：将操作系统分成两大部分
 - 运行在核心态的内核
 - 运行在用户态并以C/S方式活动的进程；





微内核结构

- 每个进程实现一类服务，称**服务器进程**
 - 如：文件服务、进程管理服务、存储管理服务、网络通信服务等。
- 任务
 - 检查有否客户提出服务请求，如果有则在满足客户的要求后返回结果，于是，用户进程与服务器进程形成了C/S关系
- 过程
 - 运行在核心态的内核把消息传给服务器；
 - 服务器执行相应操作，通过内核用消息把结果返回给用户
- 内核实现极少任务，主要起信息验证、交换的作用，因而，称**微内核 (Microkernel)**，这种OS结构也就称为客户/服务器与微内核结构



微内核结构的优缺点

- **微内核结构**将操作系统中的内存管理、设备管理、文件管理等高级服务功能尽可能从内核分离出来，变成几个独立的非内核模块
- 而在内核只保留少数最基本的功能，如最小的进程管理、最小的内存管理、进程间通信等，使内核变得简洁可靠。
- 降低了开发难度，具有较好的扩展性及移植性，特别适合大规模开放式的分布系统



微内核结构优缺点

- 缺点：效率较低
 - Windows NT 性能差于Windows95
 - Windows NT 4.0, 将部分层从用户空间移到内核空间, 实现紧耦合
 - Windows XP时, 已看不出来是微内核了。
- 几种典型的微内核OS
 - Mach: CMU, 区别UNIX的Trap机制, 引入了PORT机制, 使用IPC进行通讯, 并首次引入线程概念。后续版本应用于GNU Hurd和Mac OS X。
 - L4微内核系列: 减少IPC通讯消耗
 - Minix: Andrew S. Tanenbaum, 类Unix系统, 促成了Linux诞生, 并有Tanenbaum-Torvalds Debate
(http://groups.google.com/group/comp.os.minix/browse_frm/thread/c25870d7a41696d2)



单内核与微内核

- 操作系统内核按照其运行的情况可分为：
 - **单内核**：也称宏内核，在运行过程中，它是一个独立的进程。模块结构、层次结构的系统内核基本都是宏内核。Linux系统属于单内核类型。
 - **微内核**：微内核中，大部分内核模块都作为独立的进程，它们之间通过消息通信，模块之间互相提供服务。微内核本身类似一个消息管理器，通过合理组织内核模块来保证，只调入最需要的模块运行。Windows NT/2000系统属于改良的微内核类型。

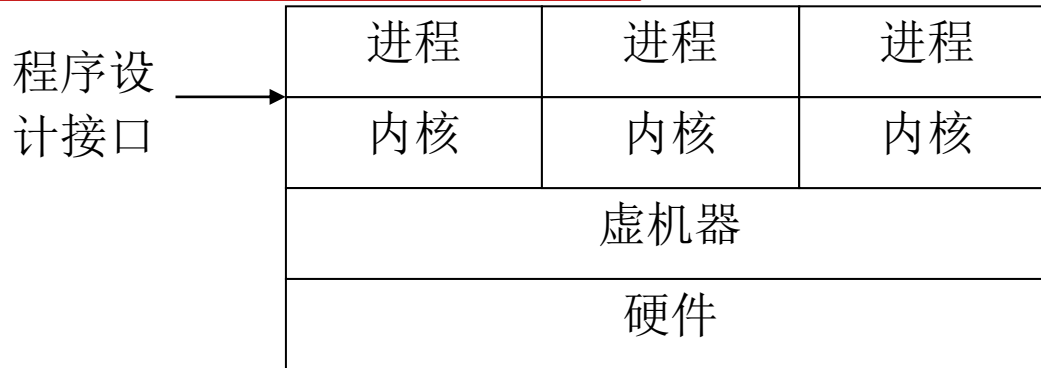


4、虚拟机结构

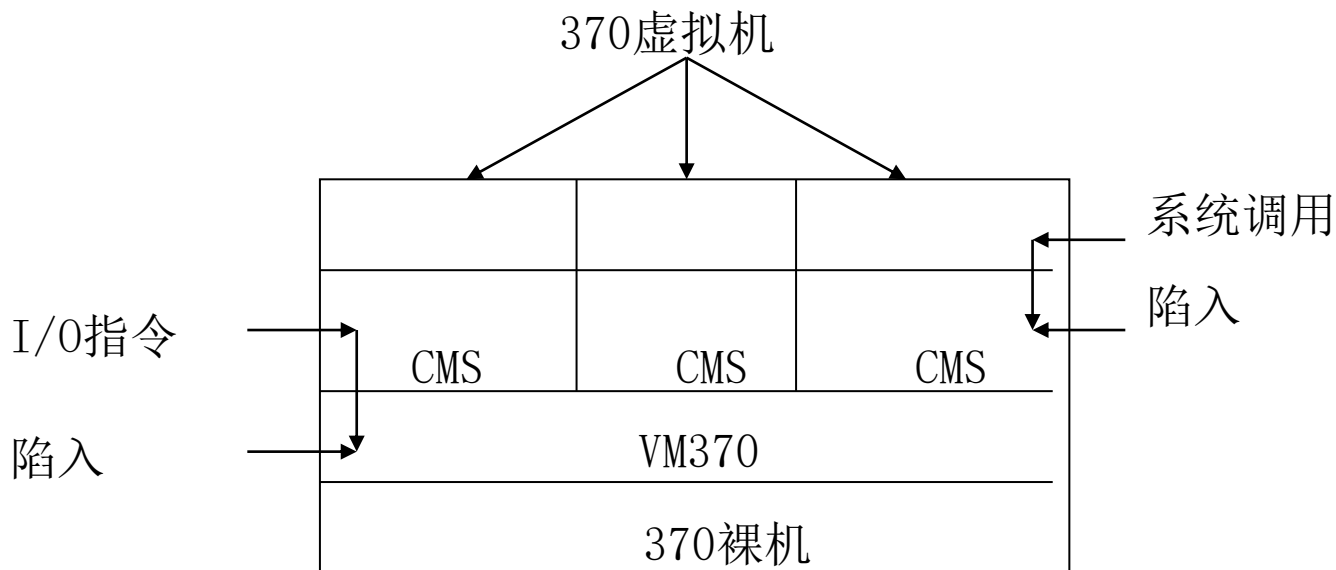
- 物理计算机资源通过多重化和共享技术可改造成多个虚拟机。
- 基本做法：**通过用一类物理设备来模拟另一类物理设备**，或通过分时地使用一类物理设备，**把一个物理实体改变成若干个逻辑上的对应物**。物理实体是实际存在的，而逻辑上的对应物是虚幻的、感觉上的。



虚拟机结构的操作系统示例



虚拟机概念结构



运行CMS (Conversational Monitor System) 的VM370虚机器



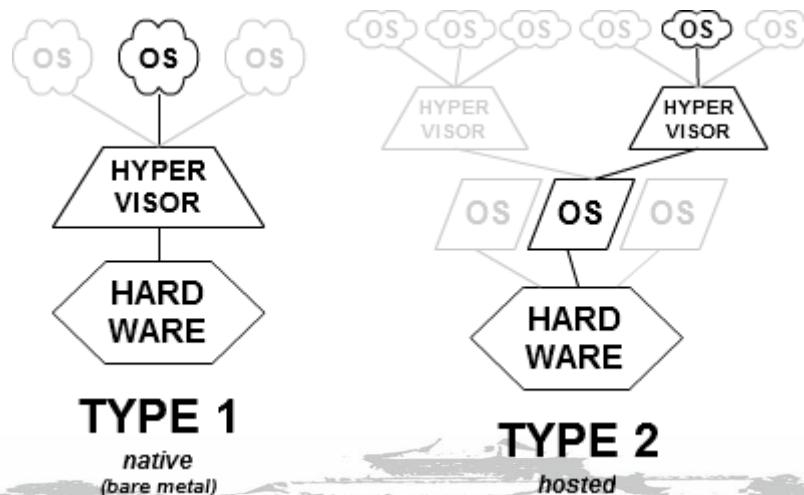
4、虚拟机结构

- Virtual Machine Monitor 或 Hypervisor
 - 为虚拟机结构提供的专门控制程序，隐藏特定计算平台的实际物理特性，为用户提供抽象的、统一的、模拟的计算环境
- Guest OS
 - 虚拟机中运行的操作系统称为客户机操作系统
 - 承载Guest OS的实体称为虚拟机，VM
- Host OS
 - 运行虚拟机监控器的操作系统称为主机操作系统
 - 某些虚拟机监控器可以脱离操作系统直接运行在硬件之上（如VMWARE 的 ESX）



虚拟机结构模式

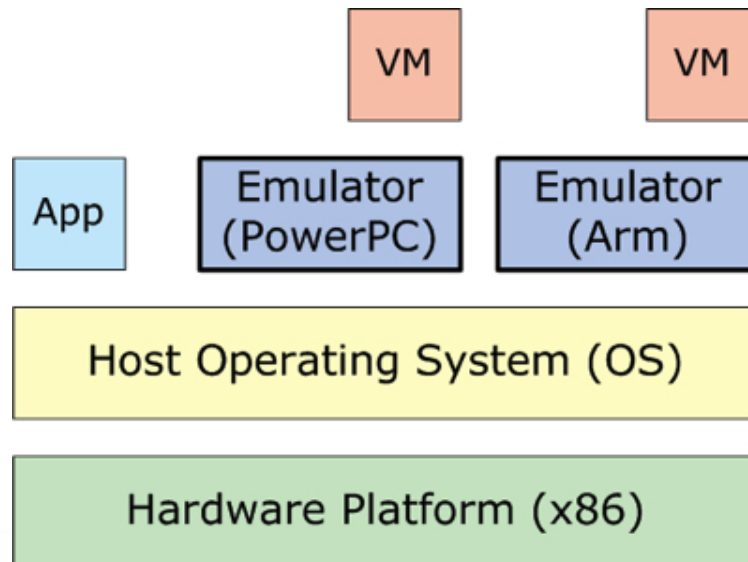
- 本地虚拟化 (native virtualization)
 - hypervisor直接运行在硬件之上，在hypervisor之上是虚拟机。(Type1)
- 主机虚拟化 (hosted virtualization)
 - hypervisor运行在操作系统之上，在同一平台上允许两个或更多的操作系统共存。(Type2)





模拟的概念

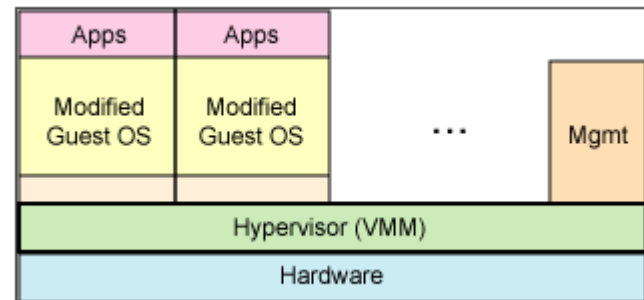
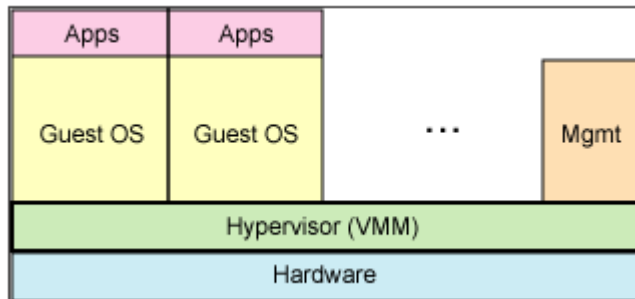
- 模拟 (Emulation) ,
 - 将一个操作系统 (Host) 的服务转换并显示成另一个操作系统 (Guest) 的过程
 - Host和Guest系统不一定是相同的, 如: Host系统可能是x86平台, 可以提供 PowerPC平台的模拟, 即使是指令和架构完全不同。





另外两种虚拟化形式

- 全虚拟化（full-virtualization）
 - 不需要修改Guest OS本身
 - 在硬件缺乏虚拟化硬件指令辅助、或者VMM不具备加速能力下，性能较低
- 半虚拟化（para-virtualization）
 - 需要修改Guest OS，添加专门虚拟化辅助指令
 - VMM不需要专门捕获和处理特权指令，性能接近物理机





硬件虚拟化

- 为了提高虚拟化性能，硬件厂商内置了相应的指令集
 - Intel VT-x
 - AMD AMD-V
 - 优点：
 - 通过引入硬件技术，将使虚拟化技术更接近物理机的速度。
 - 缺点：
 - 现有的硬件实现优化依然不足，还有进一步提高的空间。



操作系统设计的一个原则

- 机制Mechanism与策略Policy分离原则
 - 操作系统的系统调用抽象并实现特定目的和功能的函数，即**提供机制**；
 - 如何使用函数与内核无关，只由上层应用决定，即**选择策略**。
 - 提供机制的部分可以作为系统可信软件来实现；
 - 如何使用这些功能，则可在不可信环境中来定义策略。
 - 保障了将来能够最大限度调整操作系统

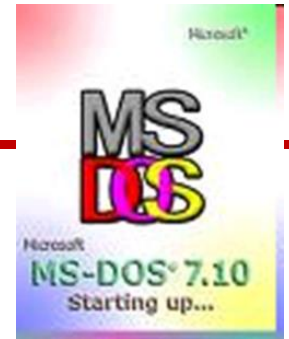


1.7 几种典型操作系统简介

- DOS 操作系统
- UNIX操作系统
- Minix操作系统
- Open Euler操作系统



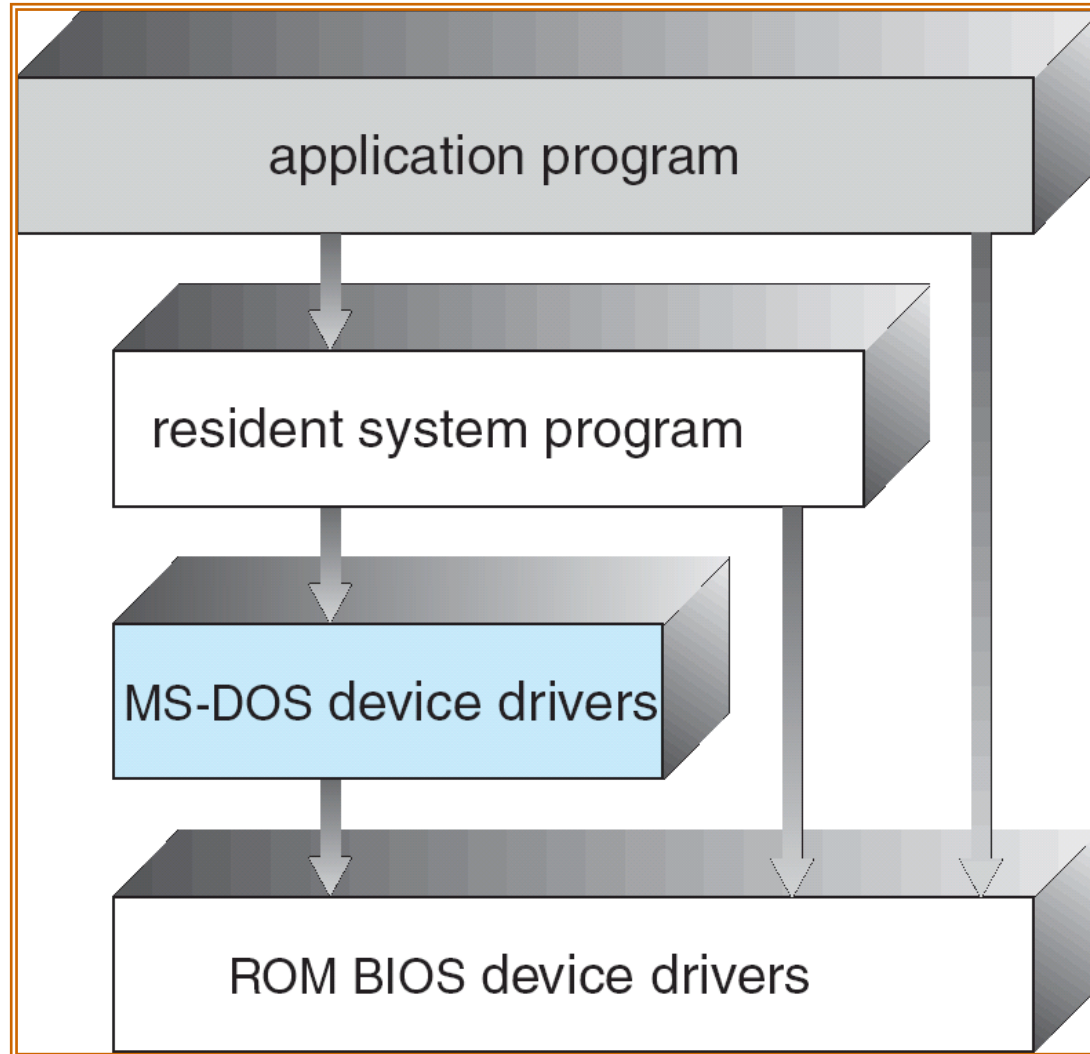
MS-DOS 操作系统



- MS-DOS：单用户单任务的操作系统
- DOS的历史：
 - 始于1981年的1.0版，由微软采购自CP/M-80变体系统86-DOS，IBM命名为PC-DOS，至1994年的最后版本DOS 6.22。DOS的主要功能有：命令处理、文件管理、设备管理、及简单的CPU调度和内存管理
- DOS的主要特点：系统开销小，运行效率高，适用于微型机
- DOS的主要缺点：无法发挥硬件能力，缺乏对数据库、网络通信的支持，没有通用的应用程序接口，用户界面不友善

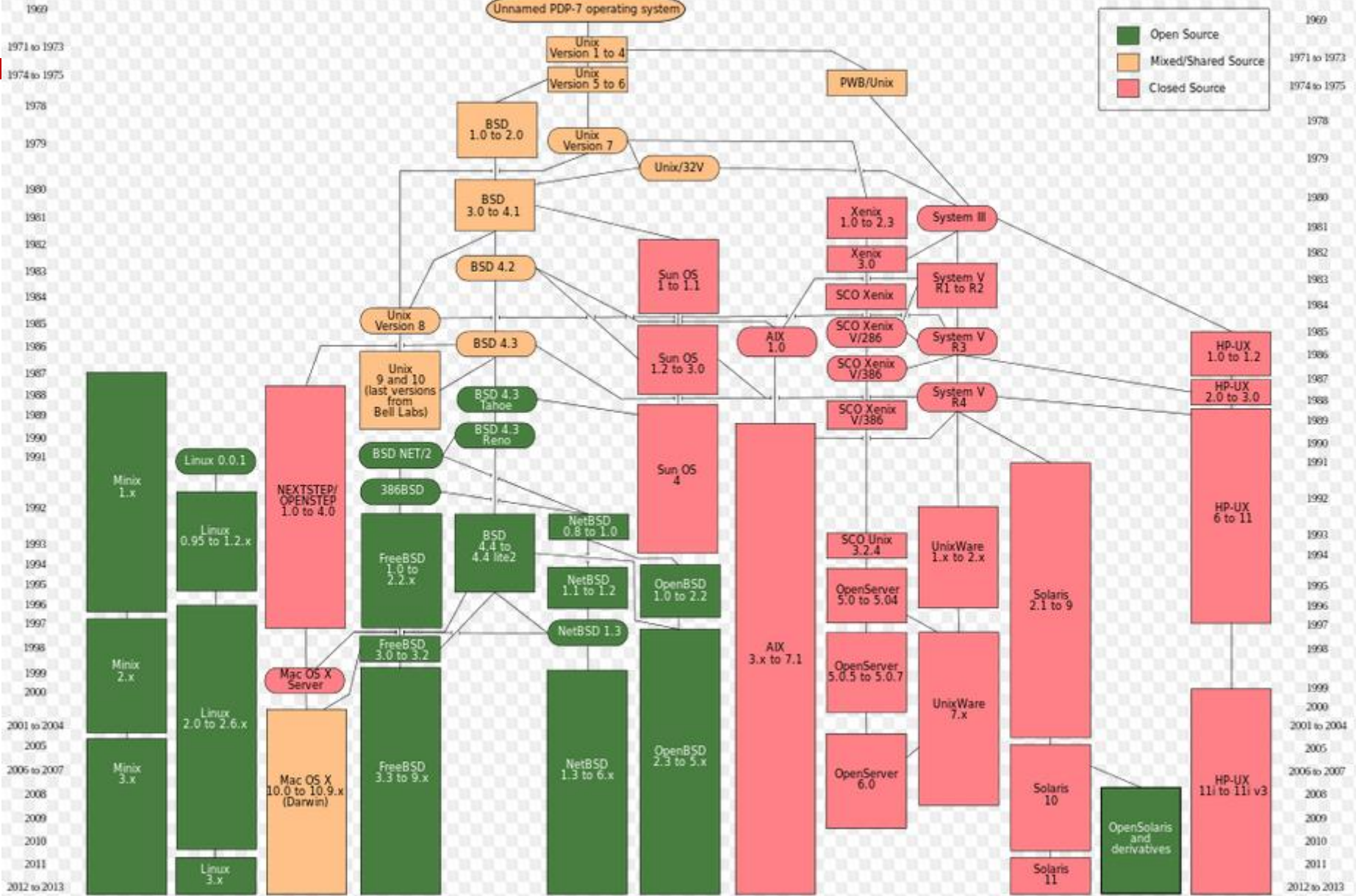


MS-DOS 操作系统





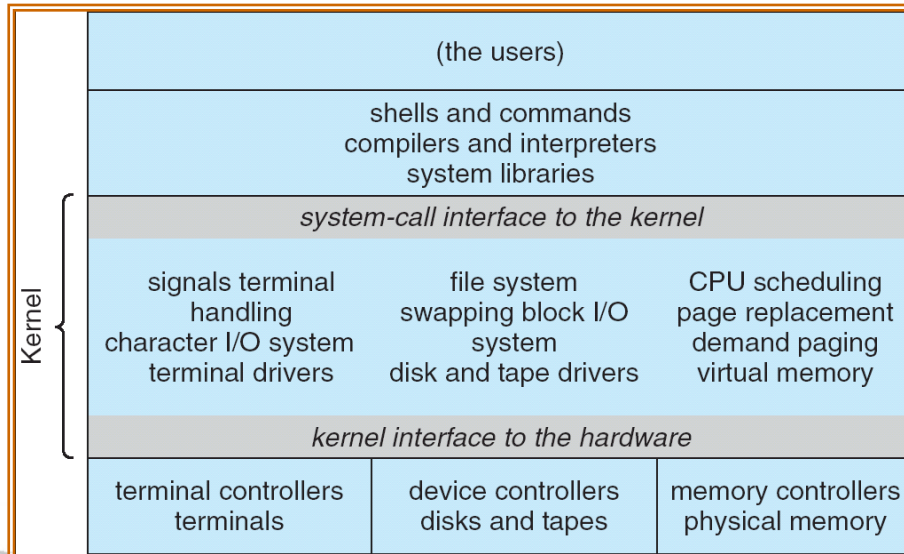
- UNIX(Uniplexed Information and Computing System)
- 诞生于1969年，至今已有50多年的历史。
- 它已从一个非常简单的操作系统发展成为性能先进、功能强大、使用广泛的操作系统，并成为事实上的多用户、多任务操作系统标准。





UNIX系统的内核结构

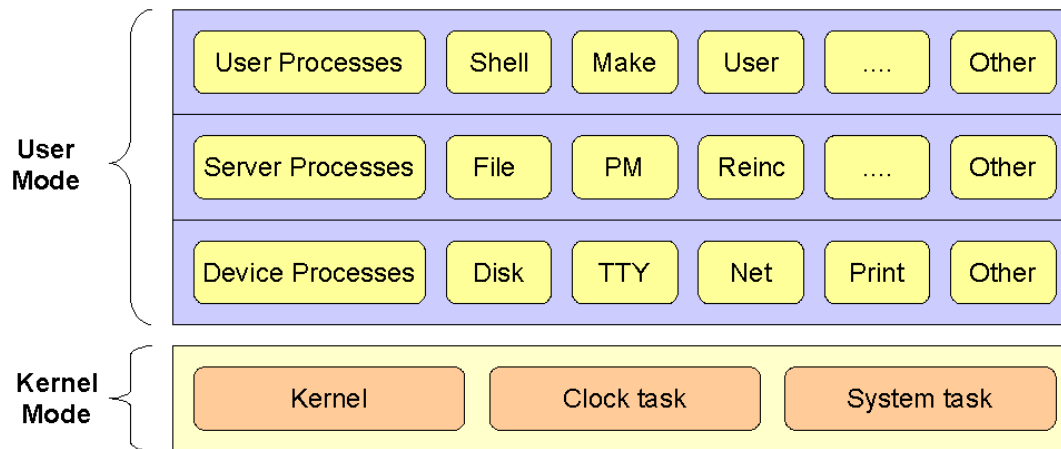
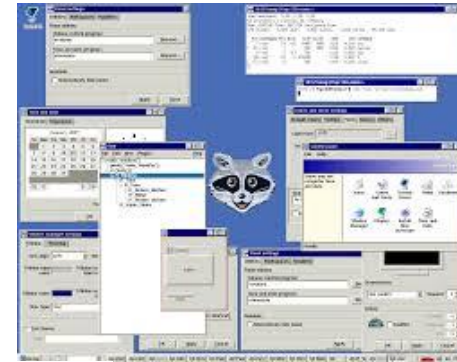
- UNIX OS包含两部分
 - 系统程序
 - 内核
 - 位于系统调用接口之下，硬件资源之上
 - 提供了文件系统、CPU调度、内存管理和其他系统功能，所有的功能都由一层来提供





MINIX操作系统

- Andrew S. Tanenbaum 开发
- 与UNIX兼容
- 教学用途
- 采用微内核结构



The MINIX 3 Microkernel Architecture



openEuler

openEuler 是一个开源、免费的Linux发行平台；
支持x86、ARM、RISC-V等多种处理器架构；
所有开发者、企业、商业组织都可以使用openEuler社区版本，
也可以基于社区版本发布自己二次开发的操作系统版本。

哪里能找到？

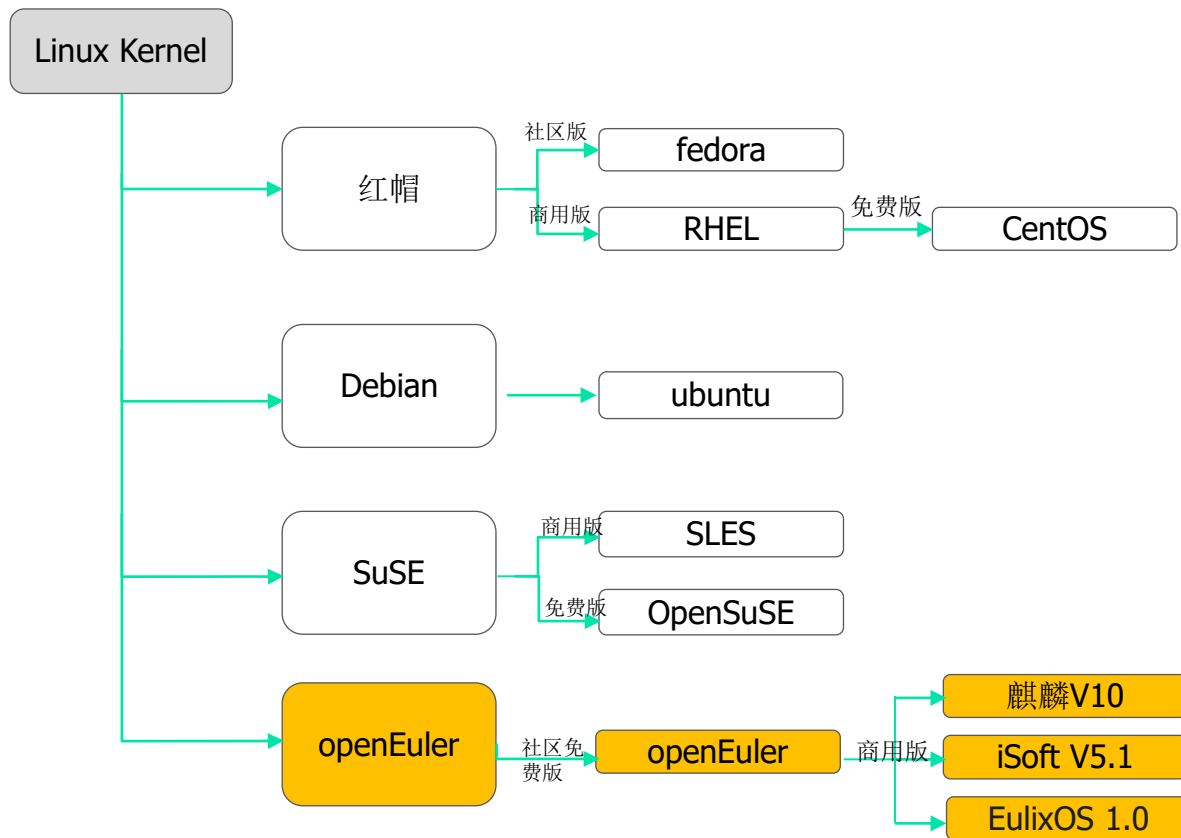
<https://openeuler.org/>

<https://gitee.com/openeuler/>





openEuler和主流OS系的关系



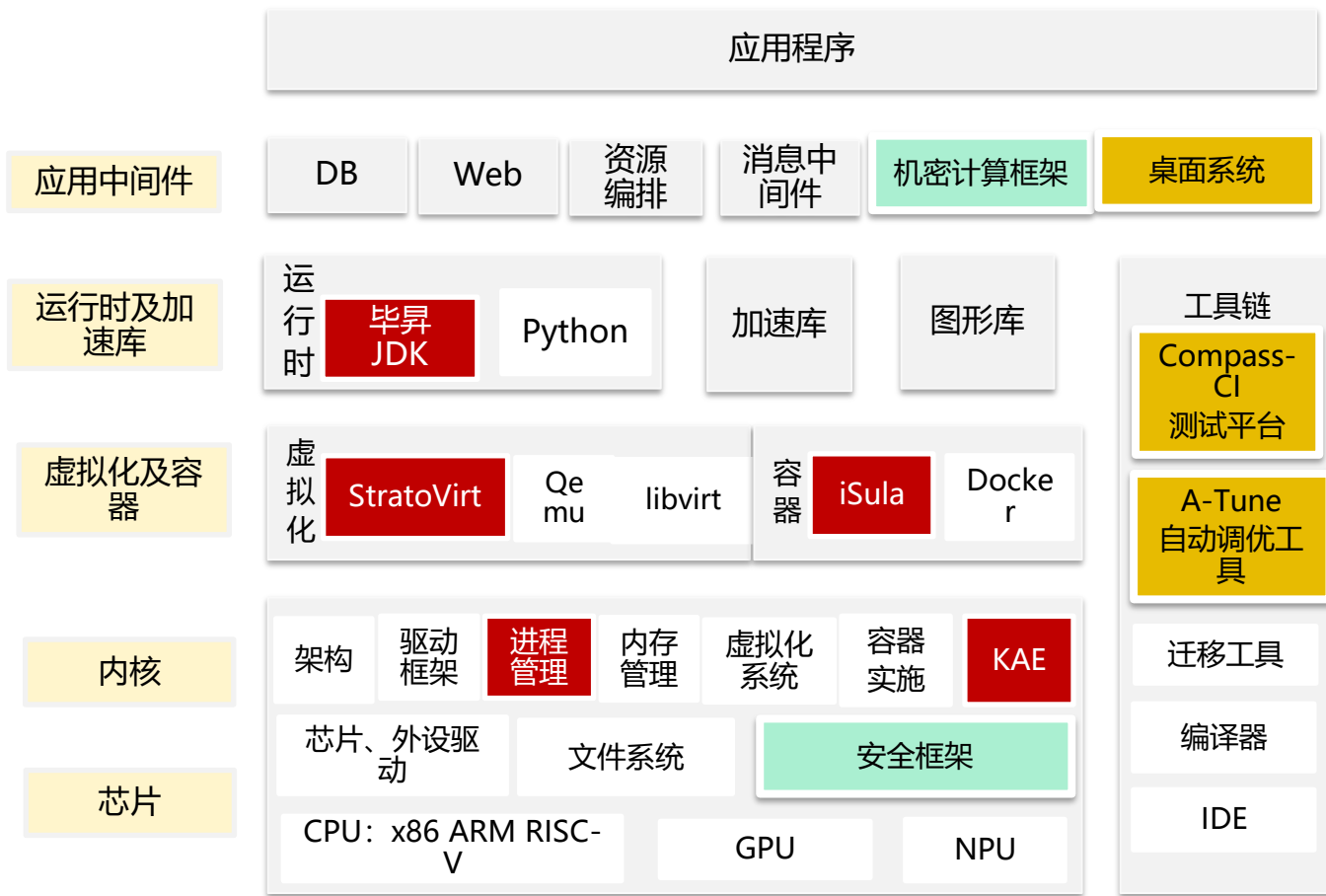
openEuler与SuSE、Debian、RedHat的上游社区都是

kernel社区www.kernel.org

openEuler社区发行LTS免费版本，使能OSV发展商业发行版，如麒麟软件、普华、中科软、万里开源等



openEuler平台框架



释放算力:

- Kernel: 分预调 +15%
- StratoVirt轻量级虚拟机: 开销 -80%, 启动速度 +10倍
- iSula 2.0云原生容器: 空载资源消耗 -68%
- 毕昇JDK: SpecJbb +20%

安全可靠:

- IMA 完整性度量架构: 防止恶意篡改
- secGear 机密计算框架: 多平台安全应用开发效率倍级提升

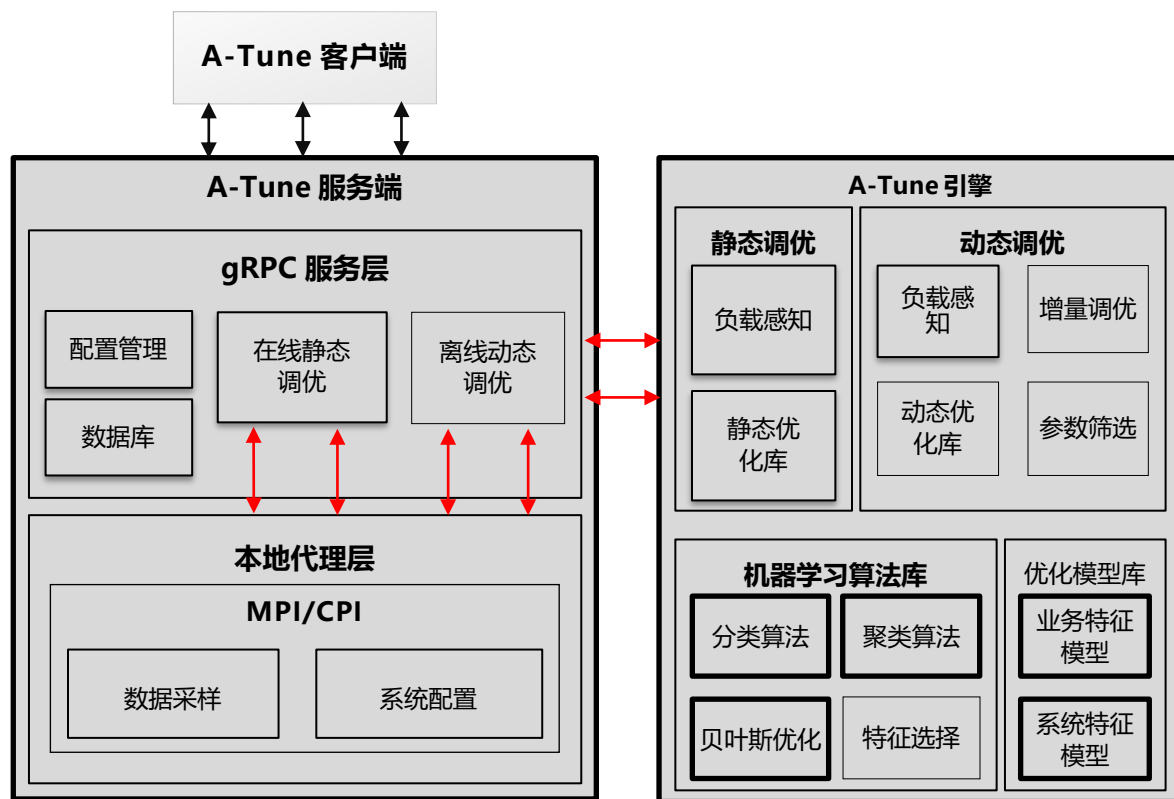
繁荣生态:

- Compass-Cl 开源软件自动化测试平台: 1000+开源软件自动化测试
- A-Tune 智能调优: 10 大类场景, 20+款应用
- UKUI 桌面: 轻量级Linux桌面环境



A-Tune 2.0

A-Tune工作模式

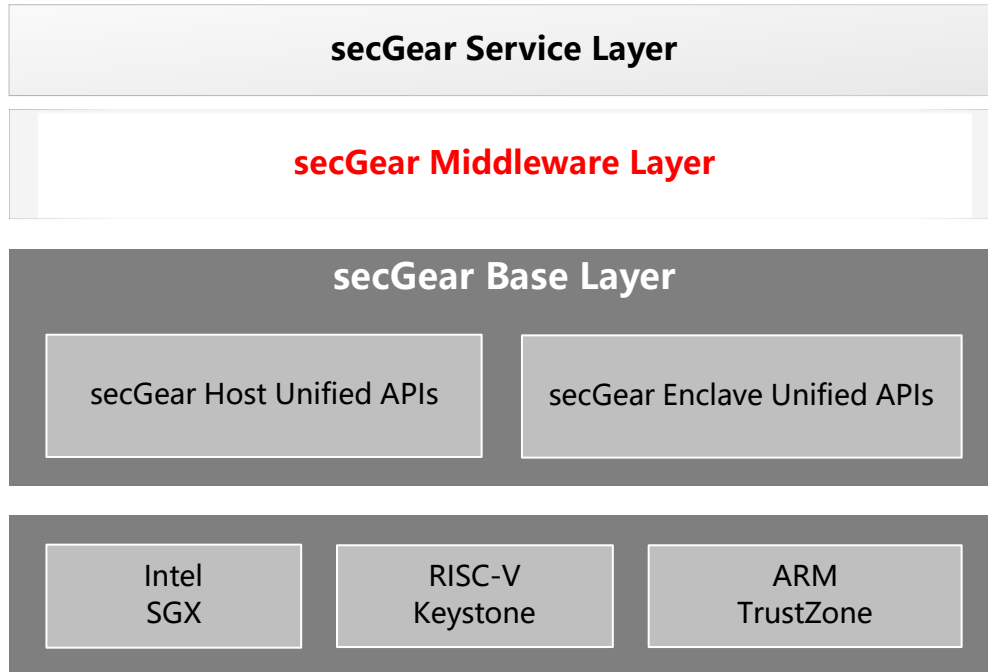


- 新增多种调优算法支持
- 支持增量调优
- 新一代负载分类模型
- **敏感参数识别与自动筛选**
- 支持虚拟机中运行
- 支持引擎独立部署
- 支持自动化模型训练



secGear

secGear机密计算统一编程框架

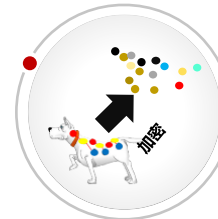


secGear Framwork



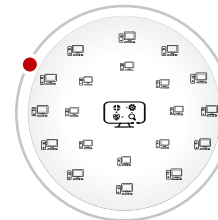
安全保障

为用户提供enclave增强的安全能力接口



丰富的服务组件

提供多种安全服务组件能力：
pkcs#11、PAKE、TLS、KMS



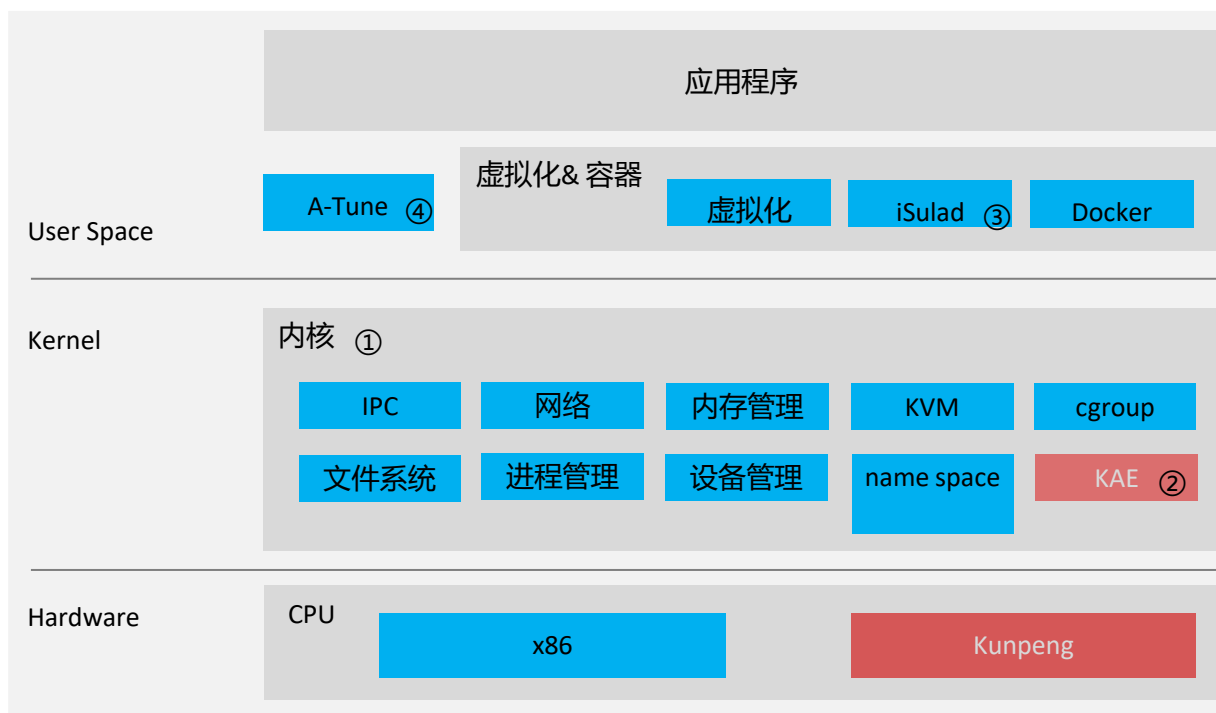
极简开发

代码生成工具生成中间代码
具备Openssl & C Posix接口库



openEuler的小结

为充分发挥鲲鹏处理器的优势，openEuler对通用Linux操作系统作了增强。



目前，openEuler所做的增强包含了如下几个方面：

1. 多核调度技术： Numa aware解决方案
2. 软硬件协同： 提供 KAE (Kunpeng Accelerator Engine) 引擎插件
3. 轻量级虚拟化： 提供iSulad 轻量级容器全场景解决方案
4. 智能优化引擎： A-Tune



本章要点回顾

1. 操作系统的地位与作用？
2. 为何需要多道程序设计？
3. 操作系统的基本类型与对应技术？
4. 操作系统的四个基本特征：并发、共享、虚拟化与异步性？
5. 操作系统的主要功能：处理器管理、存储管理、设备管理、文件管理？
6. 操作系统的接口有哪些？
7. 操作系统的内核结构有哪些？各有什么特点？模块、层次、单、微、虚拟机



本章要点回顾

- 操作系统是方便用户、管理和控制计算机软硬件资源的系统程序集合
 - 从系统角度看，OS管理计算机系统的各种资源，扩充硬件的功能，控制程序的执行
 - 从人机交互看，OS是用户与机器的接口，提供良好的人机界面，方便用户使用计算机，在整个计算机系统中具有承上启下的地位
 - 从系统结构看，OS是一个大型软件系统，其功能复杂，体系庞大，采用层次式、模块化的程序结构



小作业1

1. 在某计算机系统中，时钟中断处理程序每次执行的时间为2ms（包括进程切换开销）。若时钟中断频率为60HZ，试问CPU用于时钟中断处理的时间比率为多少？

2. 在单CPU和两台I/O(I1,I2)设备的多道程序设计环境下，同时投入三个作业运行。它们的执行轨迹如下：

Job1: I2(30ms)、CPU(10ms)、I1(30ms)、CPU(10ms)

Job2: I1(20ms)、CPU(20ms)、I2(40ms)

Job3: CPU(30ms)、I1(20ms)

如果CPU、I1和I2都能并行工作，优先级从高到低为Job1、Job2和Job3，优先级高的作业可以抢占优先级低的作业的CPU。试求：(1)每个作业从投入到完成分别所需的时间。(2)所有的作业从投入到完成CPU的利用率。(3)I/O设备利用率。



小作业1

3. 若内存中有3道程序A、B、C，它们按A、B、C优先次序运行（不考虑强占）。各程序的计算轨迹为：

A: 计算(20)、I/O(30)、计算(10)

B: 计算(40)、I/O(20)、计算(10)

C: 计算(10)、I/O(30)、计算(20)

如果三道程序都使用相同设备进行I/O(即程序用串行方式使用设备，调度开销忽略不计)。试分别画出单道和多道运行的时间关系图。两种情况下，CPU的平均利用率各为多少？



大作业1

- 教材第2章（第一部分第二章，编程项目）
 - 向OpenEuler系统，或者OpenKylin系统中，增加一个系统调用/或内核模块，鼓励程序逻辑的多样化
 - 撰写一个应用测试程序调用该系统调用/测试该内核模块
 - 使用trace/ptrace/strace,或类似的系统跟踪工具来对该测试程序进行跟踪调试
 - 附加：在指定的国产OS上，安装一个KVM，或Xen，再装一个guestOS，体会一下，或学习使用容器
- 注意事项：
 - 自由组合，每组不超过4人。
 - 提交deadline为第二章结束时，要求电子版、源码，由学委/课代表收齐给教辅。