

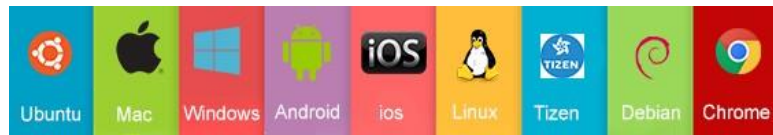
操作系统 Operating System

汤臣薇

tangchenwei@scu.edu.cn

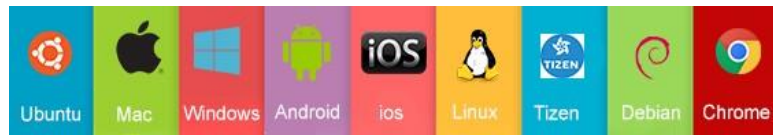
四川大学计算机学院（软件学院、智能科学与技术学院）

数据智能与计算艺术实验室



自我介绍

- **汤臣薇**，四川大学计算机学院，副教授/硕导
- **Email:** tangchenwei@scu.edu.cn
- **实验室:** 数据智能与计算艺术实验室
- **教授课程:** 操作系统原理、操作系统课程设计、数据科学引论、人工智能前沿、代码如诗：计算艺术与人文
- **研究领域:** 人工智能、深度学习、多模态学习、零样本学习、计算机视觉、工业智能、医学智能



课程基本信息

■ 面向对象

- 计算机科学与技术或相关专业

■ 课程性质

- 专业必修课，考研专业课

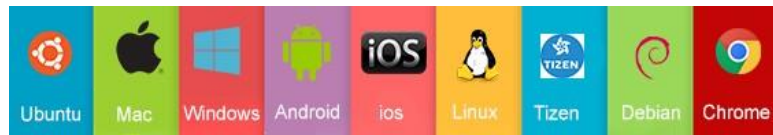
■ 操作系统原理

- 1-16周，3学分，48学时

■ 操作系统原理课程设计

- 6-15周，1学分，20学时





课程基本信息

■ 操作系统原理

- 时间：星期二 第5-7节
- 地点：江安一教A座A305

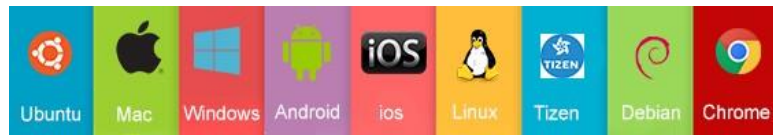
■ 操作系统课程设计

- 时间：星期二 第8-9节
- 地点：江安二基楼B301

■ 助教：启航

- 邮箱：1514894959@qq.com





课程地位和特点

■ 本课程地位

- 专业核心课程
- 考研核心课程
- 技术开发的理论基础

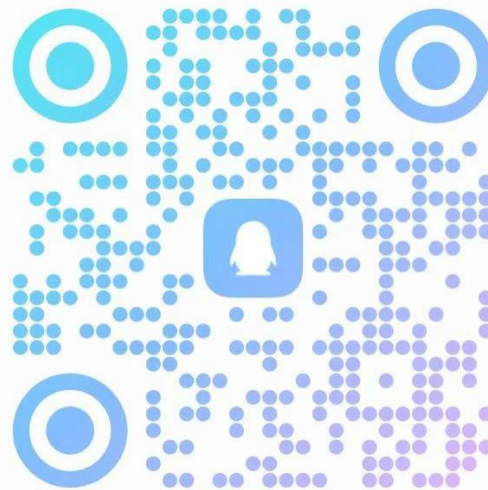
■ 课程的特点

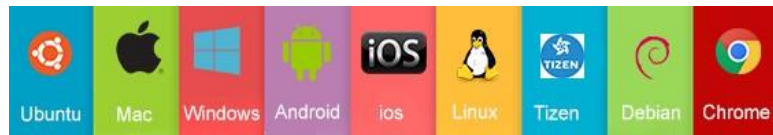
- 理论性和实践性强
- 涉及面广，错综复杂
- 抽象化，无法形象观测操作系统运行



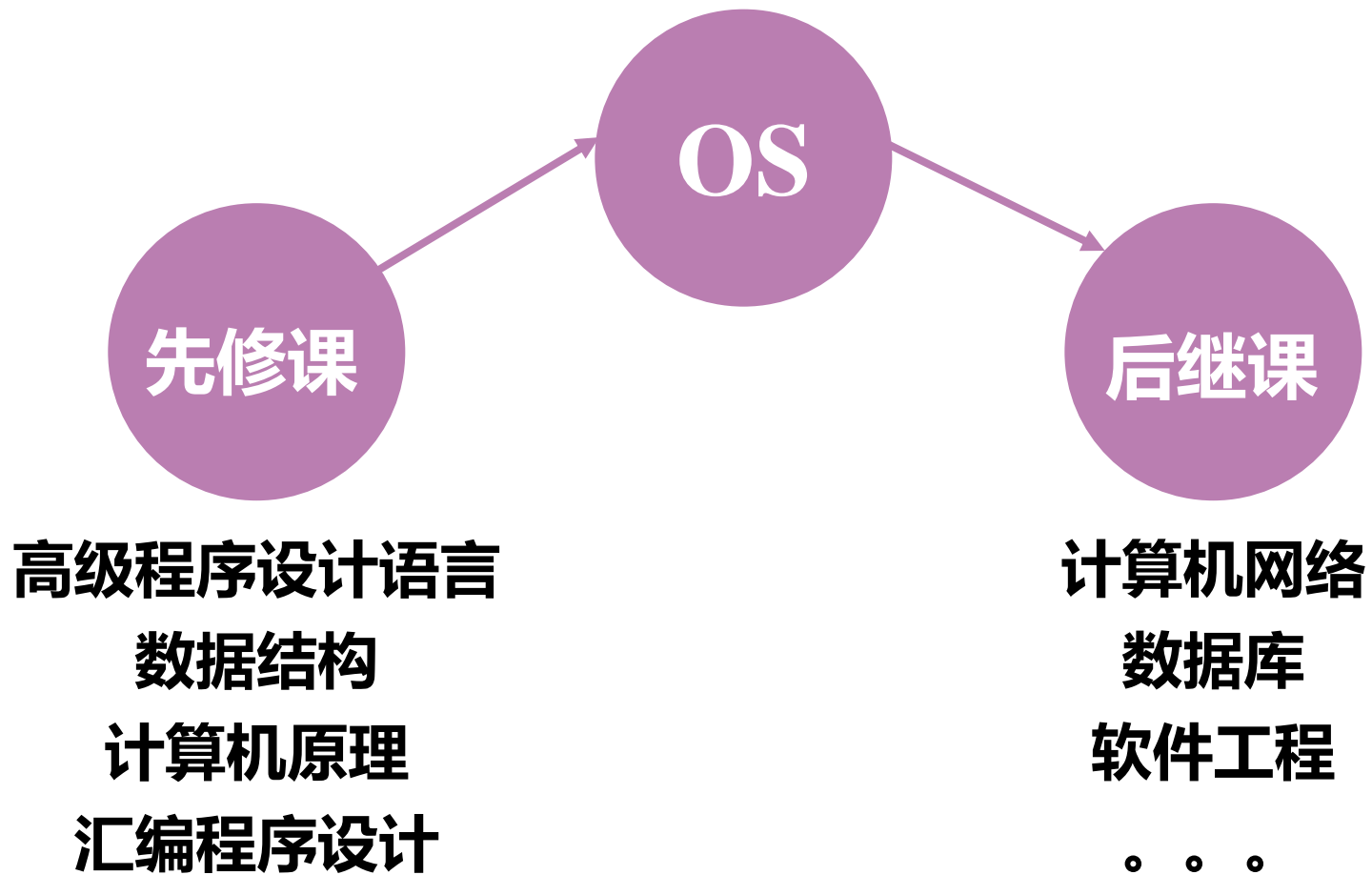
操作系统原理-2025

群号: 566815457





课程关系



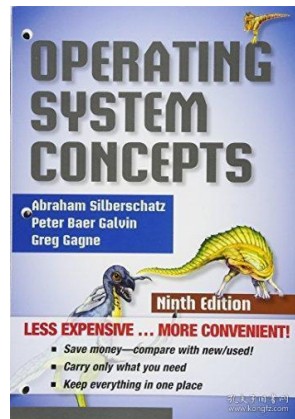
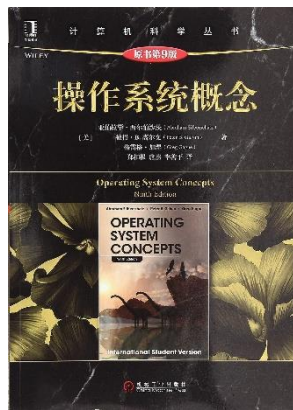
教材信息

■ 课程教材

- 汤小丹等，计算机操作系统（第四版）

■ 参考教材

- 郑扣根等，操作系统概念
- 屠祁等，操作系统基础
- 徐宗元等，操作系统
- Aadrew S.T著 陈向群等译，现代操作系统
- Abraham Silberschatz, Operation System Concepts



可借鉴的mooc资源

■ 学堂在线

- 开课学校：清华大学
- 授课老师：向勇、陈渝
- 相关网址：<https://www.xuetangx.com>



■ Coursera

- 开课学校：北京大学
- 授课老师：Professor Chen Xiangqun 陈向群教授
- 相关网址：<https://www.coursera.org/learn/os-pku#instructors>



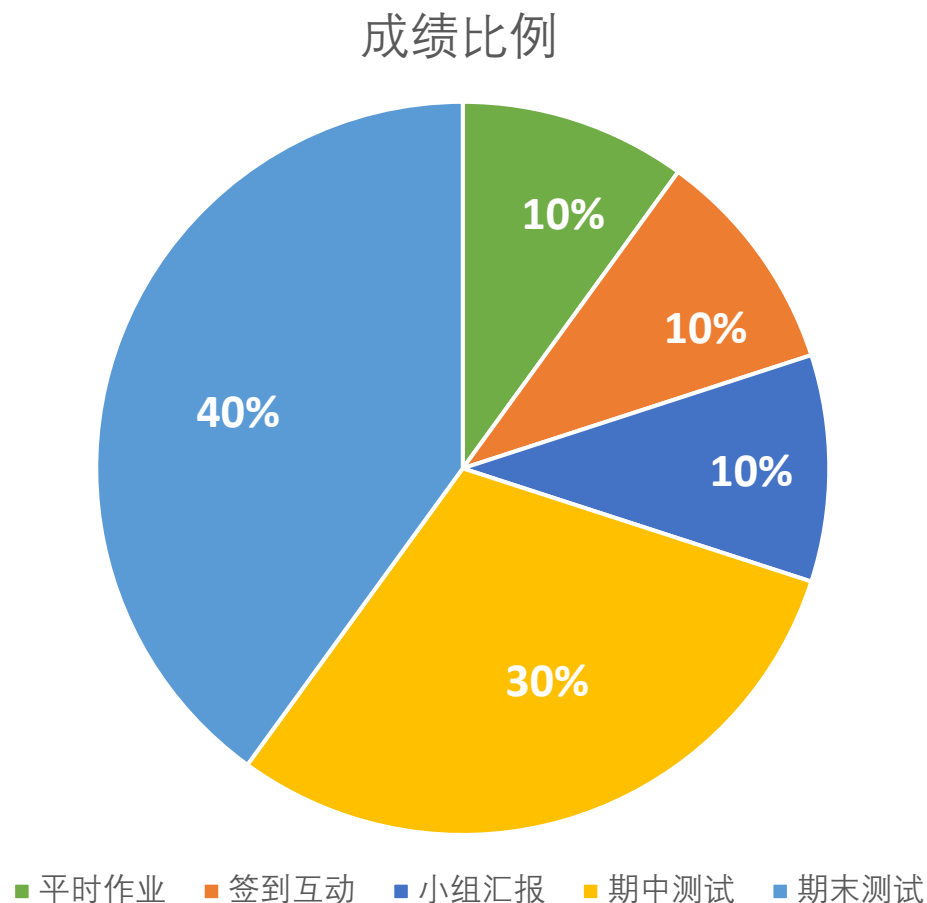
考核方式

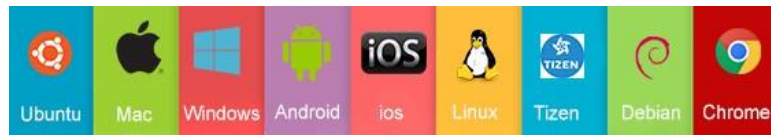
■ 平时成绩 30%

- 平时作业 10%
- 签到互动 10%
- 小组Presentation 10%

■ 考试成绩 70%

- 期中考试 30%
- 期末测试 40%





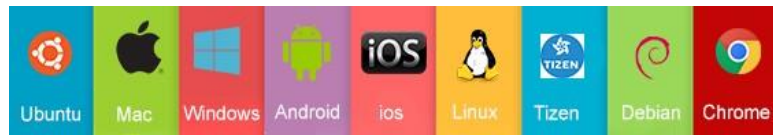
作业和报告

■ 平时作业 10%

- 练习册每周按照课程进程完成，新章课程前自行校对，期末考试当天将整本练习册上交，根据作业完成情况和校对态度进行给分
- 会随堂抽查大家的练习册完成情况

■ 小组Presentation 10%

- 期末复习时，按小组进行各章节思维导图整理和汇报



授课内容

■ 第一章 操作系统引论

➤ 1周

■ 第二章 进程的描述和控制

➤ 3周

■ 第三章 调度和死锁

➤ 2周

■ 第四章 存储器管理

➤ 1周

■ 第五章 虚拟存储器

➤ 1周

■ 第六章 输入输出系统

➤ 2周

■ 第七章 文件管理

➤ 2周

■ 第八章 磁盘存储器管理

➤ 1周

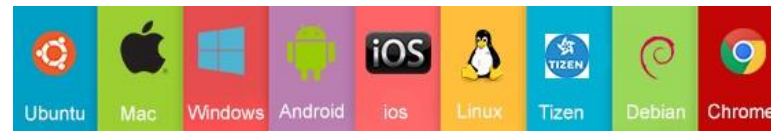
期中考试-第九周



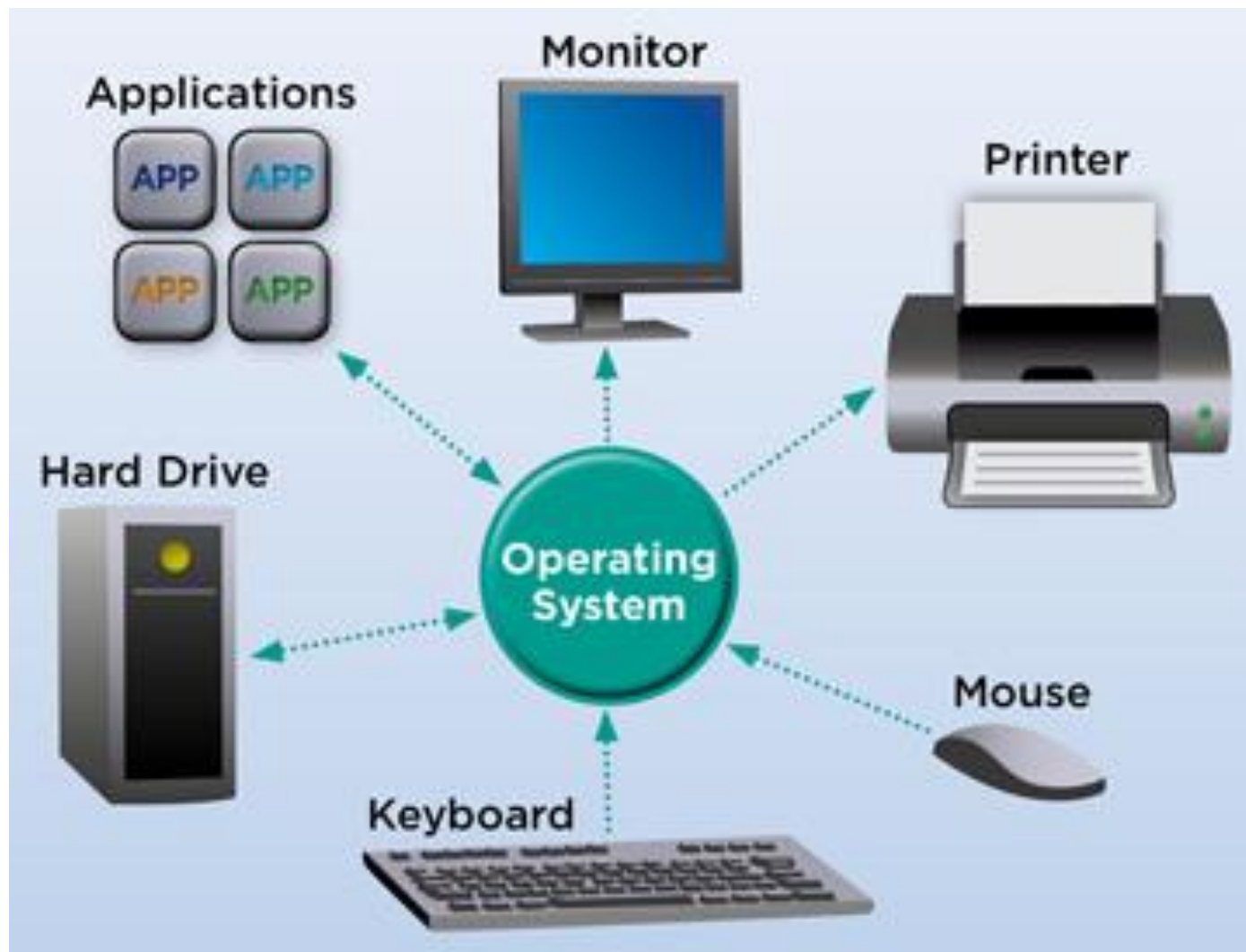
第一章

操作系统引论

操作系统引论



◆ 你知道哪些操作系统？



操作系统引论

◆ 你知道哪些操作系统？



Windows



Linux企鹅



MAC OS



Ubuntu环



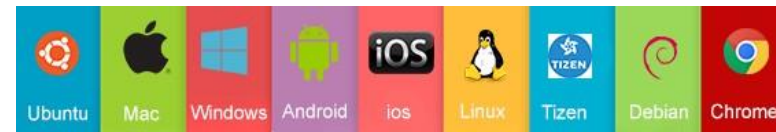
Opensuse变色龙



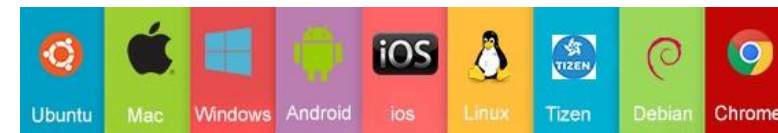
Sun的Solaris



谷歌的Android

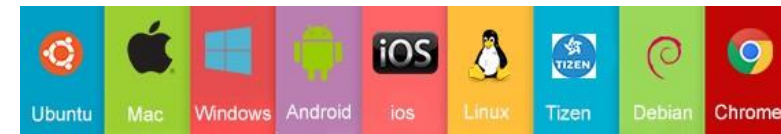


操作系统引论



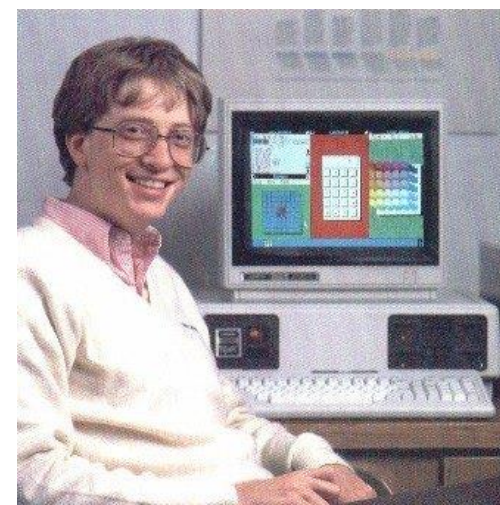
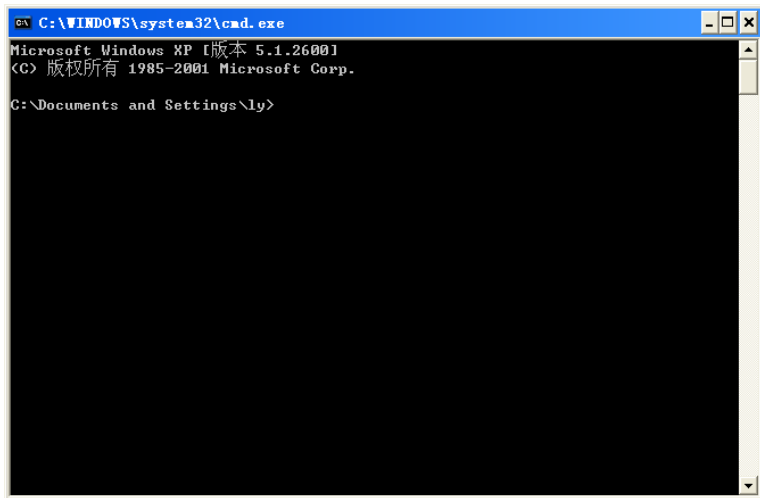
MITS电脑公司·基于Intel 8080芯片的Altair 8800微型计算机·人类历史上第一台PC

操作系统引论

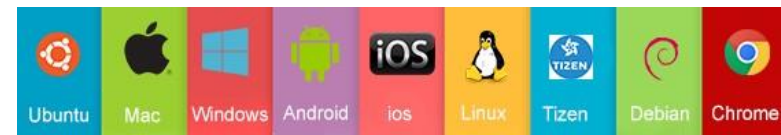


◆ DOS (Disk Operating System) 磁盘操作系统：一种PC操作系统

- 1980年，IBM公司“国际象棋”的项目：开发价格低廉的微型计算机，面向中小企业和家庭用户市场，选微软为其新PC机编写关键的操作系统；
- 微软从帕特森手中买下操作系统QDOS的使用权，部分改写后给IBM，命名为Microsoft DOS；
- IBM-PC机的普及使MS-DOS取得了巨大成功。



操作系统引论



◆ Windows XP 操作系统

- 2001年10月25日，微软推出Windows XP操作系统，比尔·盖茨宣布——“DOS时代到此结束。”

```
C:\>dir

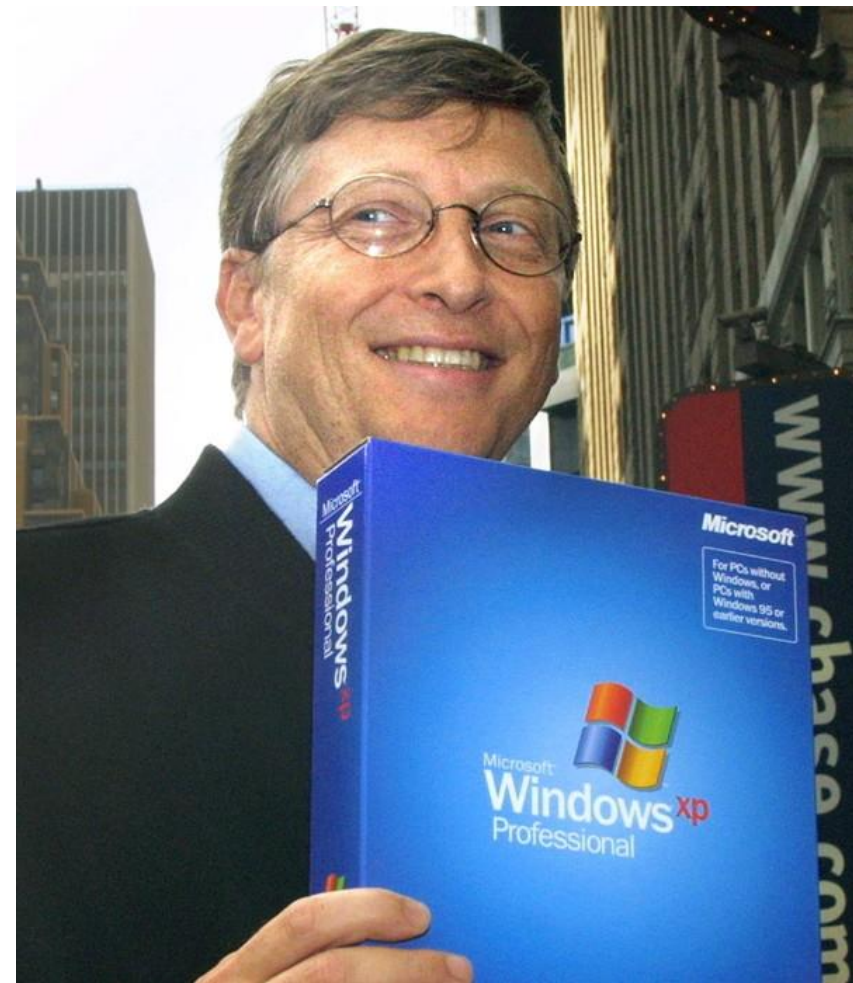
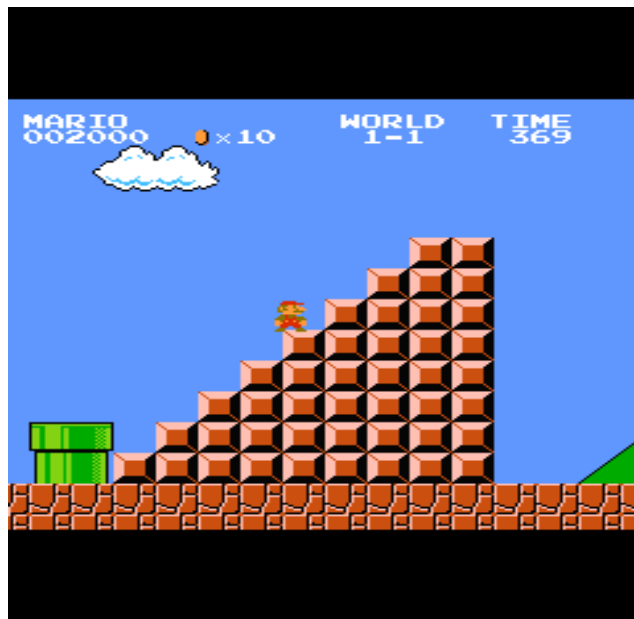
Volume in drive C is PC DISK
Volume Serial Number is 3143-BEFO
Directory of C:\

DOS             <DIR>          10-03-04   11:55p
VB DOS          <DIR>          10-04-04   12:16a
UCDOS           <DIR>          10-04-04   12:15a
UCDI CT         <DIR>          10-04-04   12:15a
WINDOWS         <DIR>          10-04-04   12:19a
VB              <DIR>          10-04-04    9:32a
               6 file(s)          0 bytes
               1,874,526,208 bytes free

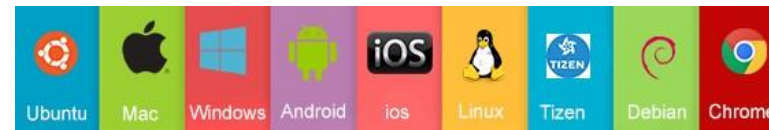
C:\>ver

MS-DOS Version 6.22

C:\>_
```



操作系统引论

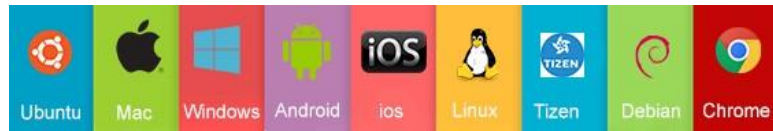


◆ Windows 系列

典型版本	日期
Windows 1.0	1985-11-20
Windows 2.0	1987-11-01
Windows 3.0	1990-05-22
Windows 3.1	1992-03-18
Windows 3.2	1994-04-14
Windows NT	1993-07-27
Windows 95	1995-08-24
Windows 98	1998-06-25
Windows 2000	2000-02-17
Windows ME	2000-09-14

典型版本	日期
Windows XP	2001-10-25
Windows Server 2003	2003-04-24
Windows Vista	2007-01-30
Windows Server 2008	2008-02-27
Windows 7	2009-10-22
Windows 8	2012-10-26
Windows 8.1	2013-10-17
Windows 10	2015-07-29
Windows 11	2021-06-24
...	...

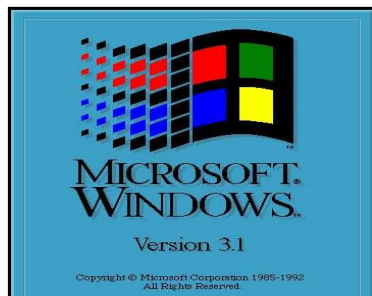
操作系统引论



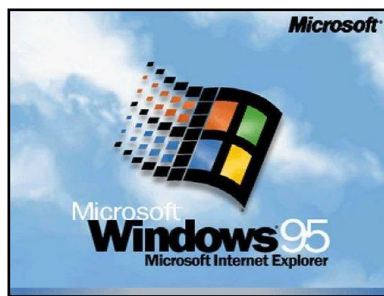
◆ Windows 系列



Window 1.0



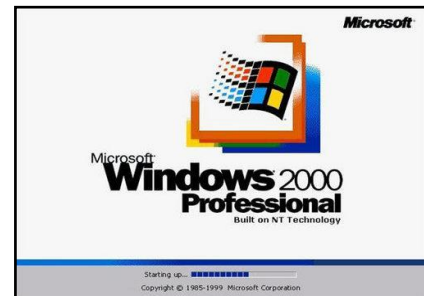
Windows 3.1



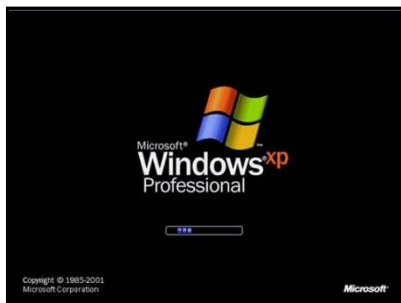
Windows 95



Windows 98



Window 2000



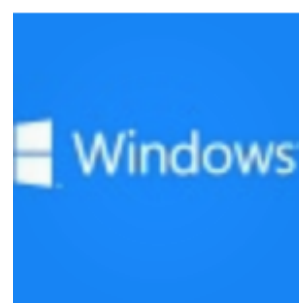
Windows XP



Windows Vista



Windows 7



Windows 8



Windows 10

操作系统引论



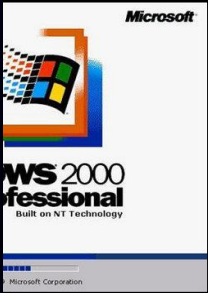
◆ Win



W



Win

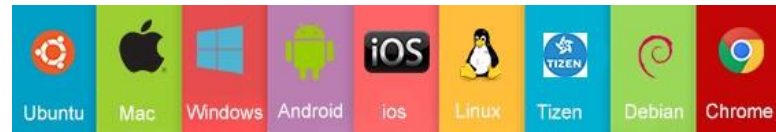


ow 2000



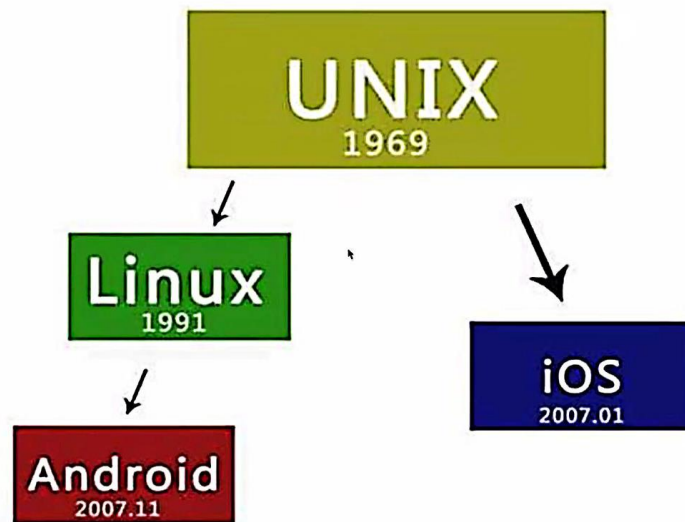
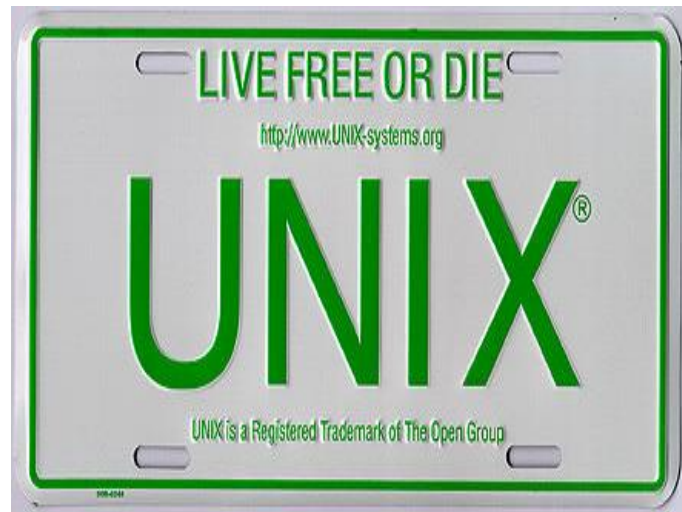
ows 10

操作系统引论

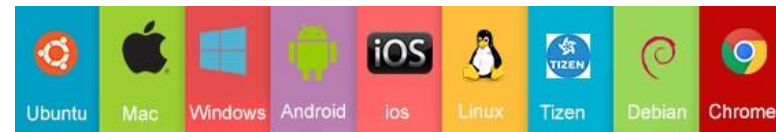


◆ Unix操作系统

- Unix是一个强大的多用户多任务操作系统。
- 它用C语言编写，支持多种处理器架构，按照操作系统的分类，属于分时操作系统，最早由1969年在AT&T的贝尔实验室开发。
- Unix是第三次工业革命中计算机软件领域最具代表性的产物。
- 在近40年中，由Unix造成的影响是最有深远意义的。



操作系统引论

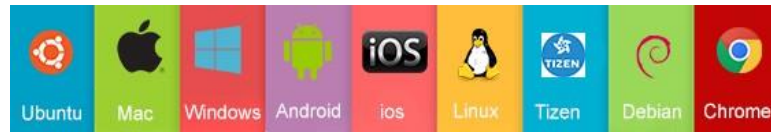


◆ Linux

- Linux是**最受欢迎的自由电脑操作系统内核**。
- 它是一个用C语言和汇编语言写成的类Unix操作系统。
- Linux最早是由芬兰黑客托瓦兹为尝试在英特尔x86架构上提供自由免费的**类Unix操作系统**而开发的。
- 主流的Linux发行版本包括：**Ubuntu**、Fedora、Red Hat、红旗等。

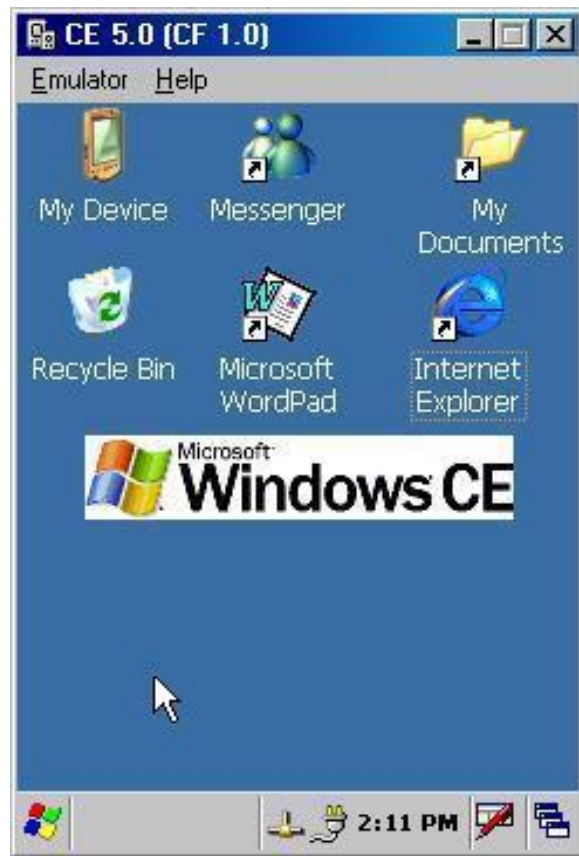
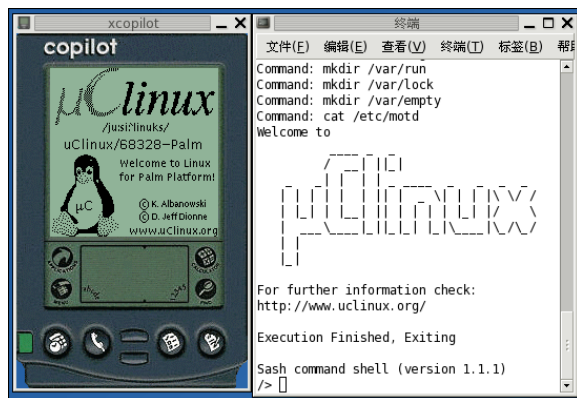


操作系统引论



◆ 嵌入式操作系统

- uClinux: micro-control linux, 即 **“微控制器领域中的Linux系统”**，主要是针对目标处理器没有存储管理单元的嵌入式系统而设计的，已经被成功地移植到了很多平台上。
- Windows CE: 微软公司嵌入式、移动计算平台的基础，它是一个开放的、可升级的**32位嵌入式操作系统**，是基于掌上型电脑类的电子设备操作系统，其图形用户界面相当出色。

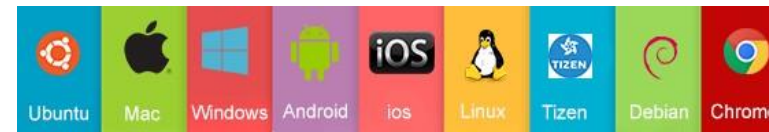


操作系统引论

◆ 嵌入式操作系统

- Symbian：塞班公司**为手机而设计的操作系统**。

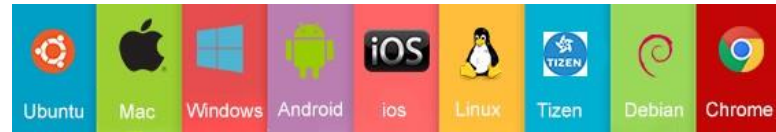
2008年12月2日，塞班公司被诺基亚收购。2011年12月21日，诺基亚官方宣布放弃塞班（Symbian）系统品牌，下一版本操作系统将更名为诺基亚Belle。截止至2011年11月，塞班系统的中国市场占有率降至23%，**被安卓超过**。



symbian
OS



操作系统引论

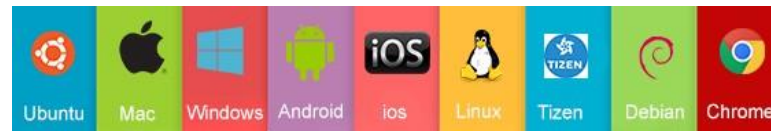


◆ 嵌入式操作系统

- Andriod（安卓）：**谷歌公司**收购安迪鲁宾的Android公司并于2007年11月5日，正式展示了Android操作系统，Android是**基于Linux开放性内核的操作系统**。它采用了软件堆层的架构。底层Linux内核只提供基本功能，其他的应用软件由各公司自行开发，部分程序以Java编写。2011年初Android超越称霸十年的塞班，跃居全球最受欢迎的智能手机平台。



操作系统引论

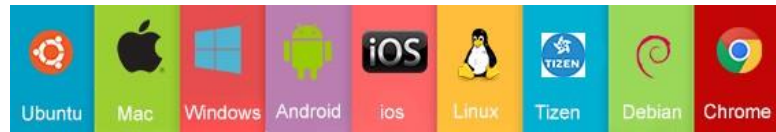


◆ 嵌入式操作系统

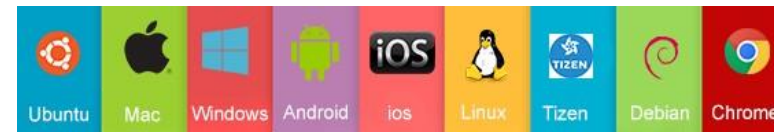
- iOS：由**苹果公司**为iPhone开发的操作系统，它主要是给iPhone、iPod touch以及iPad使用；原本这个系统名为iPhone OS，直到2010年6月7日宣布改名为iOS。
- iOS的系统架构分为四个层次：核心操作系统层、核心服务层、媒体层、可轻触层。



操作系统引论



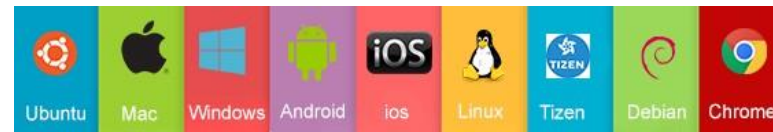
1.1 操作系统的概念



OS是计算机系统中一组**控制**和**管理**计算机硬件资源和软件**资源**，合理地各类作业进行**调度**，以**方便用户**使用的**程序的集合**

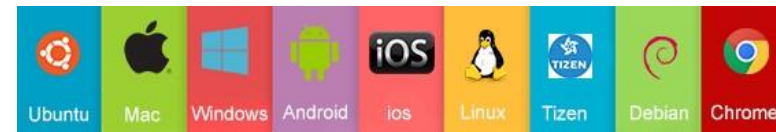


1.1 操作系统的目标



- ◆ **方便性**（用户的观点）：提供**良好的、一致的和易用的用户接口**，弥补硬件系统的类型和数量差别。使用人员不需要懂得很多计算机的知识就能使用计算机
- ◆ **有效性**（系统管理人员的观点）：管理和分配硬件、软件资源，合理地组织计算机的工作流程，**提高系统资源的利用率和吞吐量**
- ◆ **可扩充性**（系统设计员的观点）：硬件技术的发展和计算机功能和性能不断提出的更高要求，要求操作系统的设计应采用良好的设计方法和结构，以便很方便地增加新功能、改进老功能
- ◆ **开放性**：计算机网络的发展，计算机之间能够通过网络进行通讯，协同工作；同时程序的**可移植性和互操作性**也要求OS具有开放性

1.1 操作系统的作用



1. OS作为用户与计算机硬件系统之间的接口

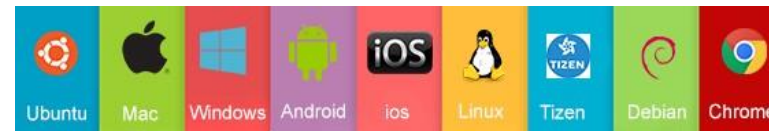
- 直接位于计算机硬件之上，为计算机应用提供接口
- 提供通用的计算机服务，与专用的应用领域无关
- 实现资源管理策略，为不同的应用提供共享资源

操作系统

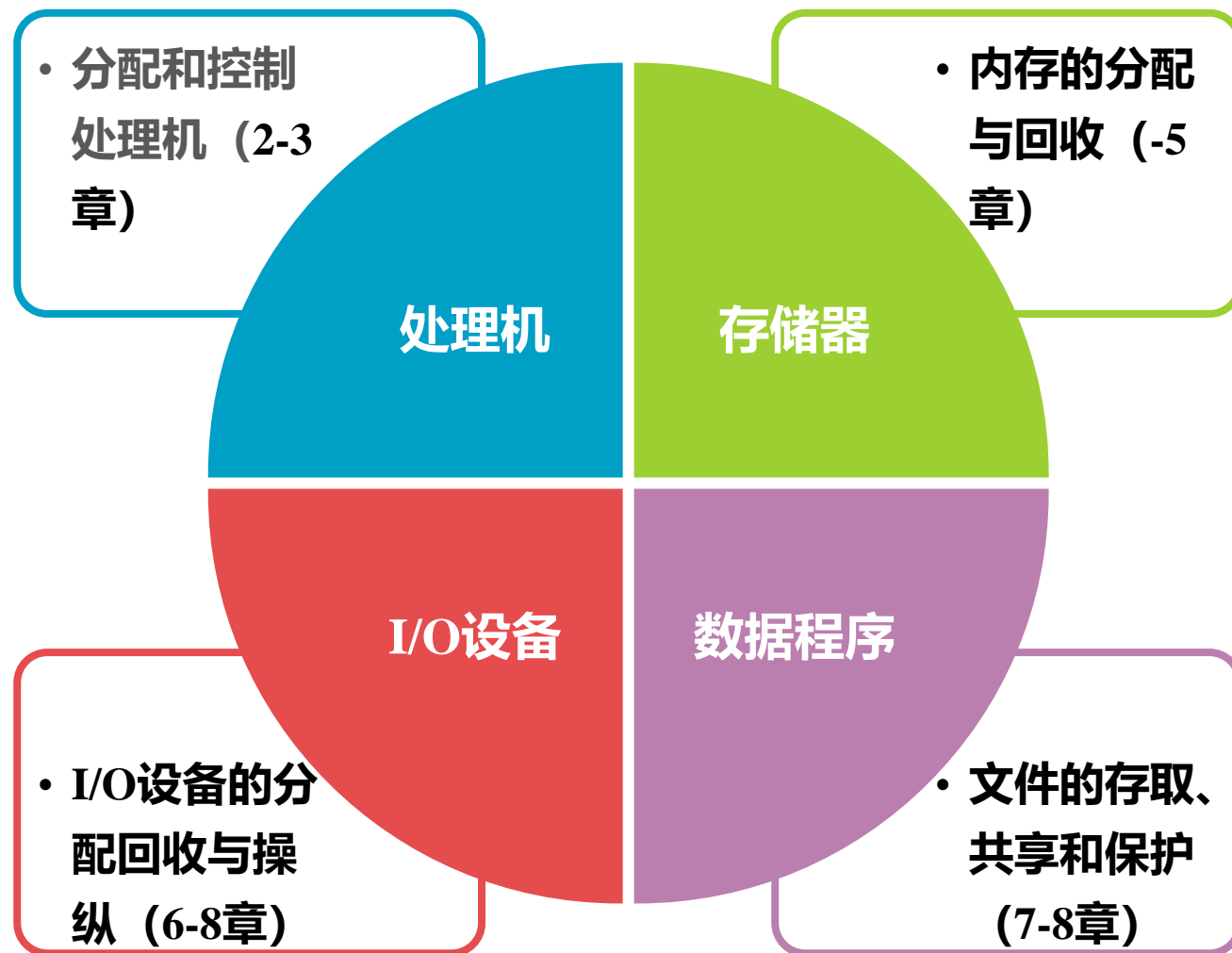
计算机硬件



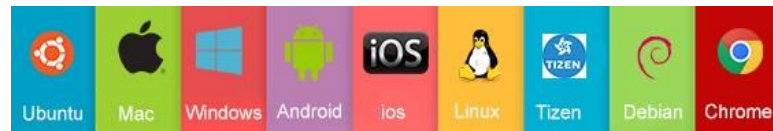
1.1 操作系统的作用



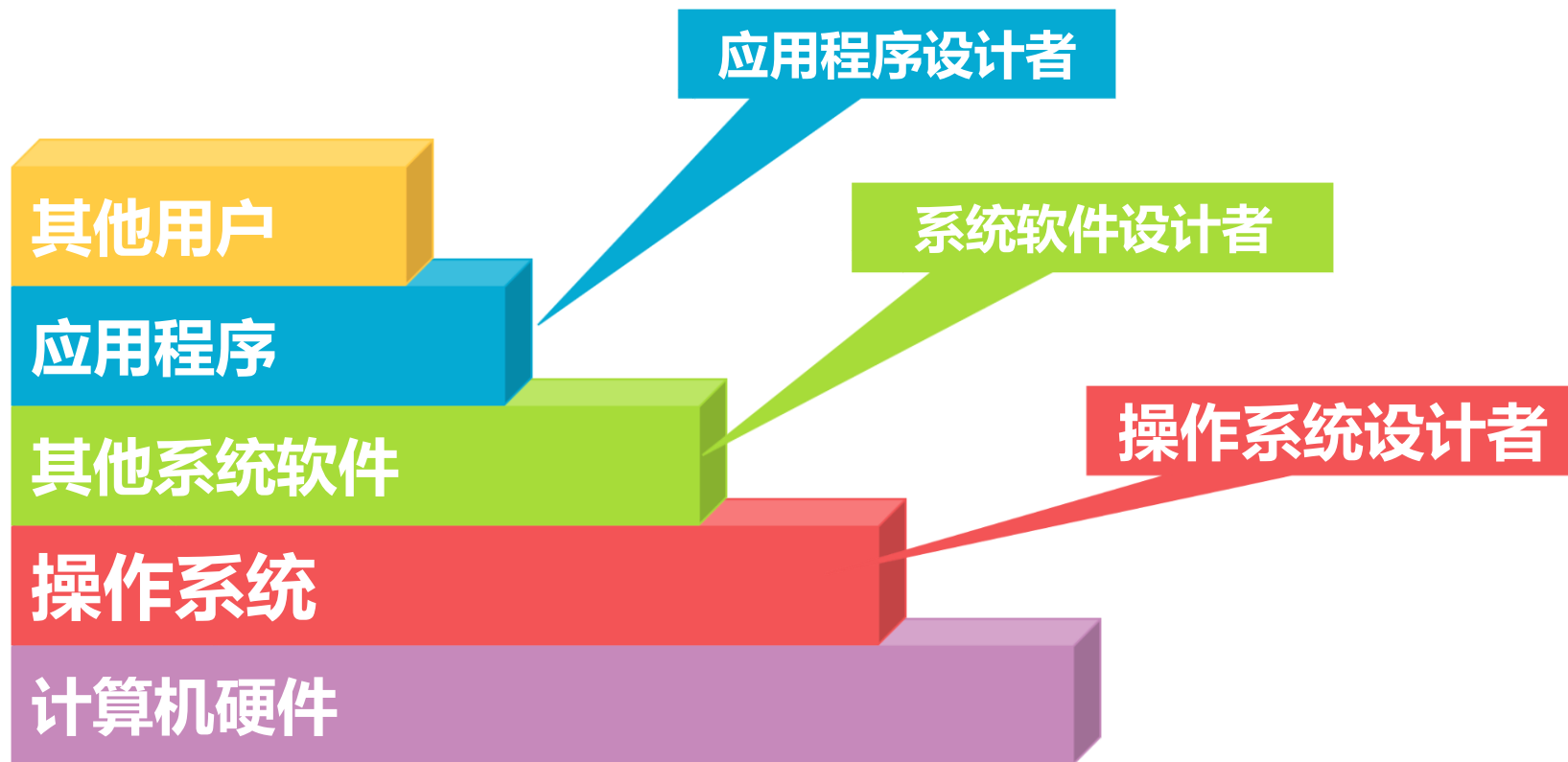
2. OS作为计算机系统资源的管理者



1.1 操作系统的作用

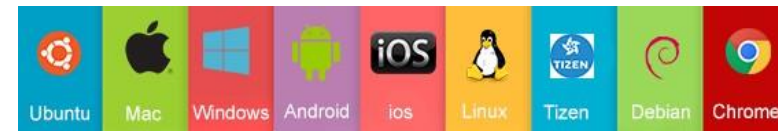


3. OS实现了对计算机资源的抽象

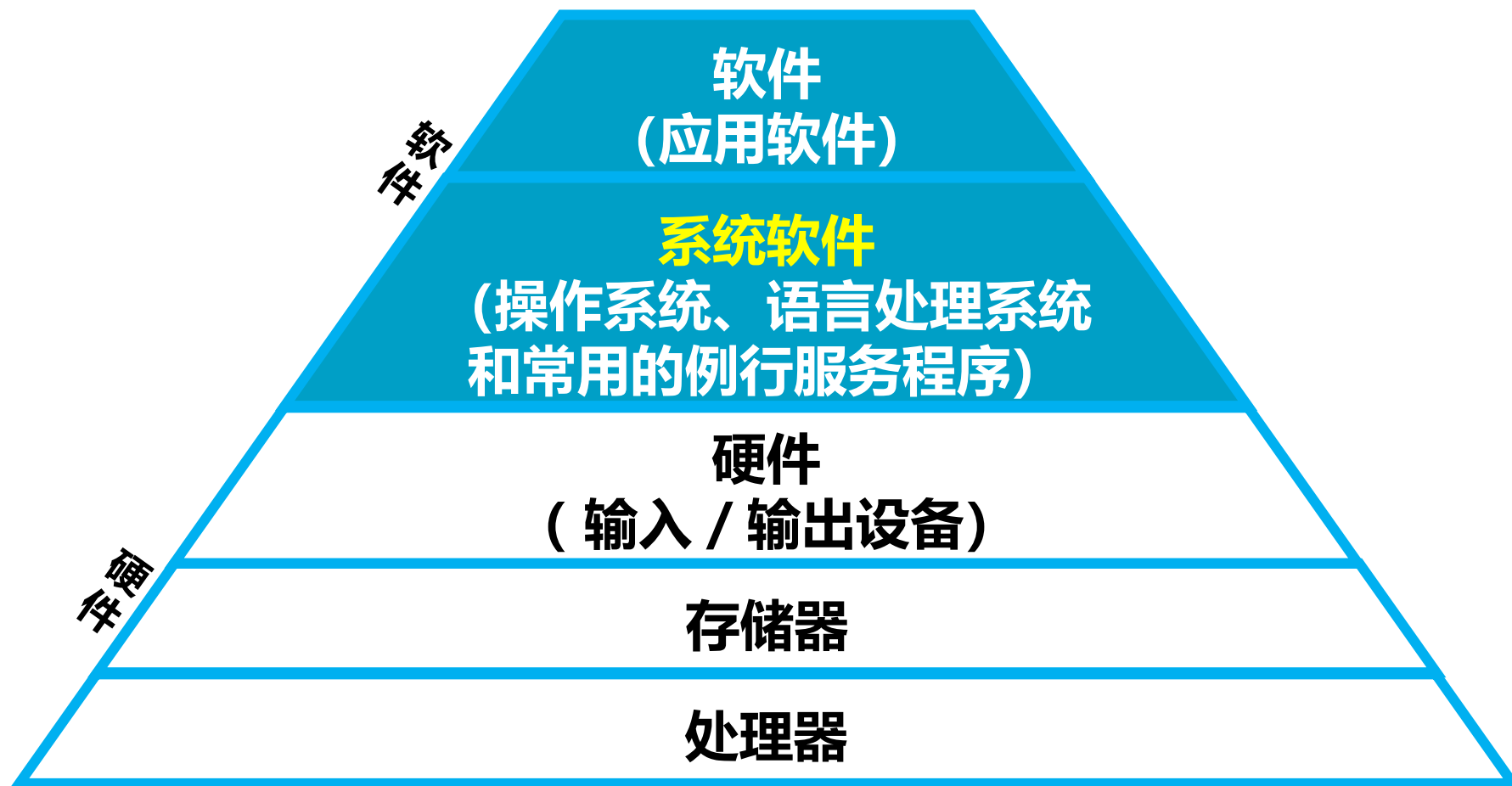


OS与开发者的关系

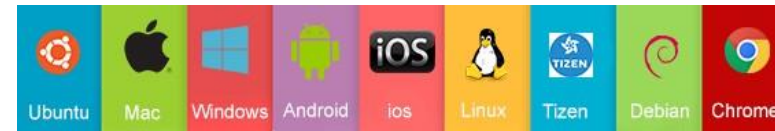
1.1 操作系统的作用



4. OS是系统软件



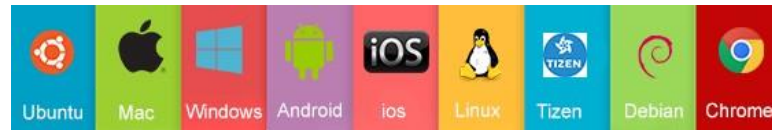
1.1 操作系统的作用



4. OS是系统软件

- **裸机**：没有配置任何软件的计算机
 - **软件**：在硬件基础之上对硬件的性能加以扩充和完善
- (虚拟机：计算机 + 软件 → 功能更强大使用更方便的机器)**
- **操作系统**是计算机的第一个软件，管理和控制计算机系统中的硬件及软件资源、合理地组织计算机工作流程，以便有效利用这个资源为用户提供一个功能强大使用方便和可扩充的工作环境，从而在计算机与用户之间起到接口作用。

1.1 操作系统的作用



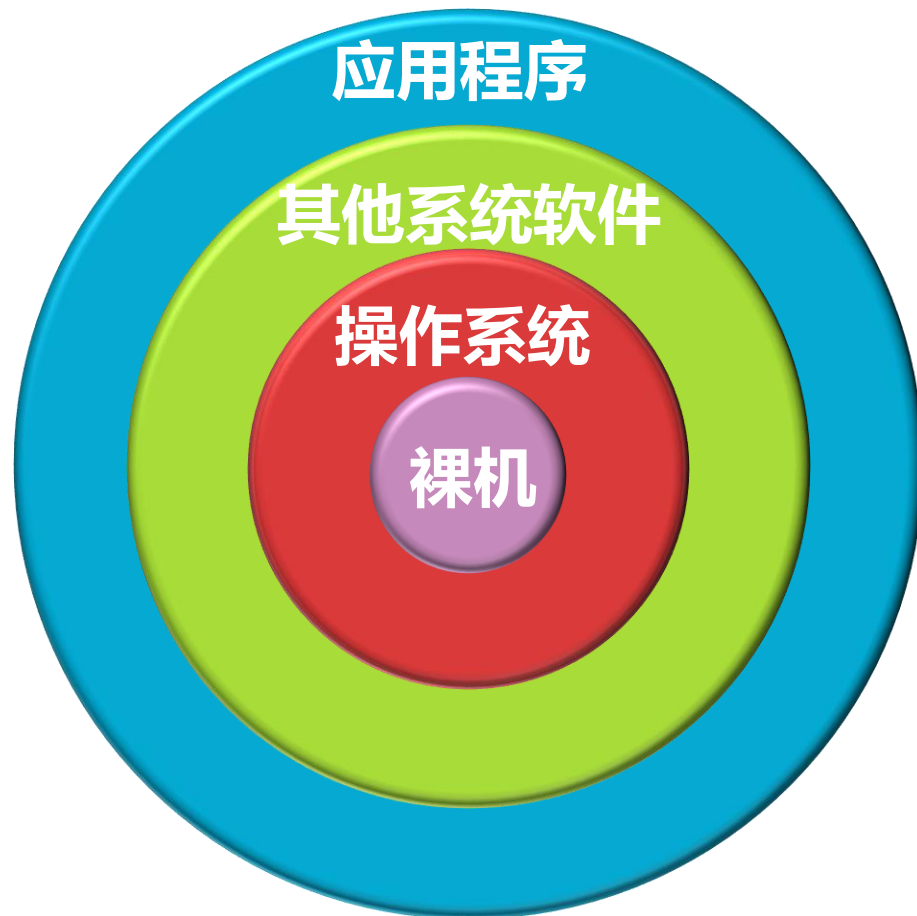
4. OS是系统软件

➤ OS与硬件的关系

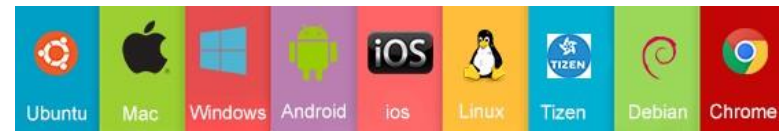
- 控制CPU的工作
- 访问存储器
- 设备驱动、中断处理

➤ OS与用户及应用程序的关系

- 提供方便的用户界面
- 提供优质的服务

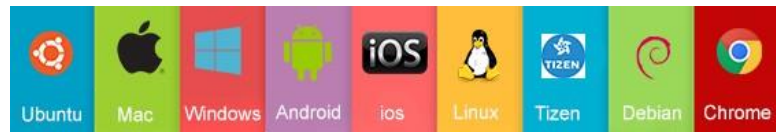


1.1 推动OS发展的动力

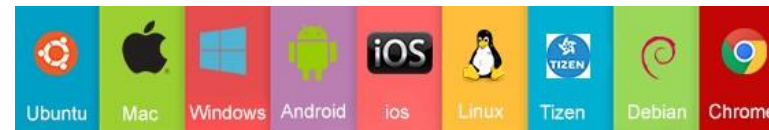


- ◆ **提高资源的利用率和系统性能**：例如从单用户系统到多用户系统，从单道系统到多道系统等
- ◆ **方便用户**：用户上机、调试程序，分散计算时的事务处理和非专业用户（商业和办公、家庭）
- ◆ **器件的发展**：CPU的位宽度（指令和数据）、快速外存
- ◆ **计算机体系结构的发展**：单机系统到多机系统，孤立的计算机系统到通过网络相连的计算机系统等，要求计算机操作系统与之相适应
- ◆ **不断提出新的应用需求**：如将计算机应用于工业控制中，需要配置能进行实时控制的OS；要求提供听音乐、看电影、玩游戏等多媒体功能；嵌入式操作系统等

操作系统引论



1.2 OS发展历史



未配置OS的计算机系统

单道批处理系统(simple batch processing)

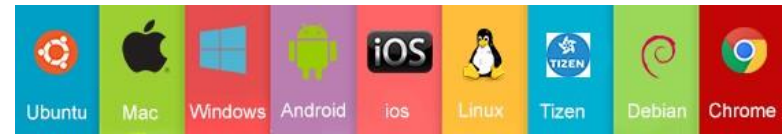
多道批处理系统(multiprogramming system)

分时系统(time-sharing system)

实时系统(real-time system)

微机操作系统(microcomputer operating system)

1.2 OS发展历史



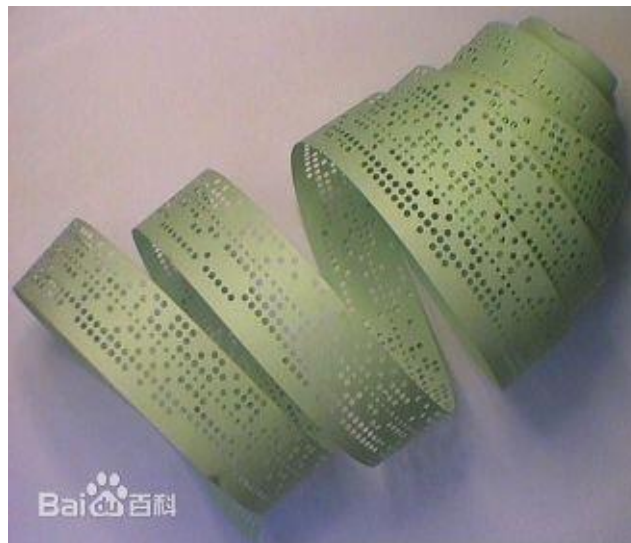
再将打好的卡纸放入卡纸阅读机中

1.2.1 未配置OS的计算机系统

◆1946 ~ 50年代中期（电子管），集中计算（计算中心），计算机资源昂贵

➤ 工作方式：人工操作

- 用户：程序员/操作员
- 编程语言：为机器语言
- 输入输出：纸带或卡片



➤ 计算机的工作特点

- 用户独占全机：不出现资源被其他用户占用，资源利用率低
- CPU等待人工操作：计算前，手工装入纸带或卡片；计算完成后，手工卸取纸带或卡片；CPU利用率低
- 每一用户都要自行编写涉及到硬件的源代码

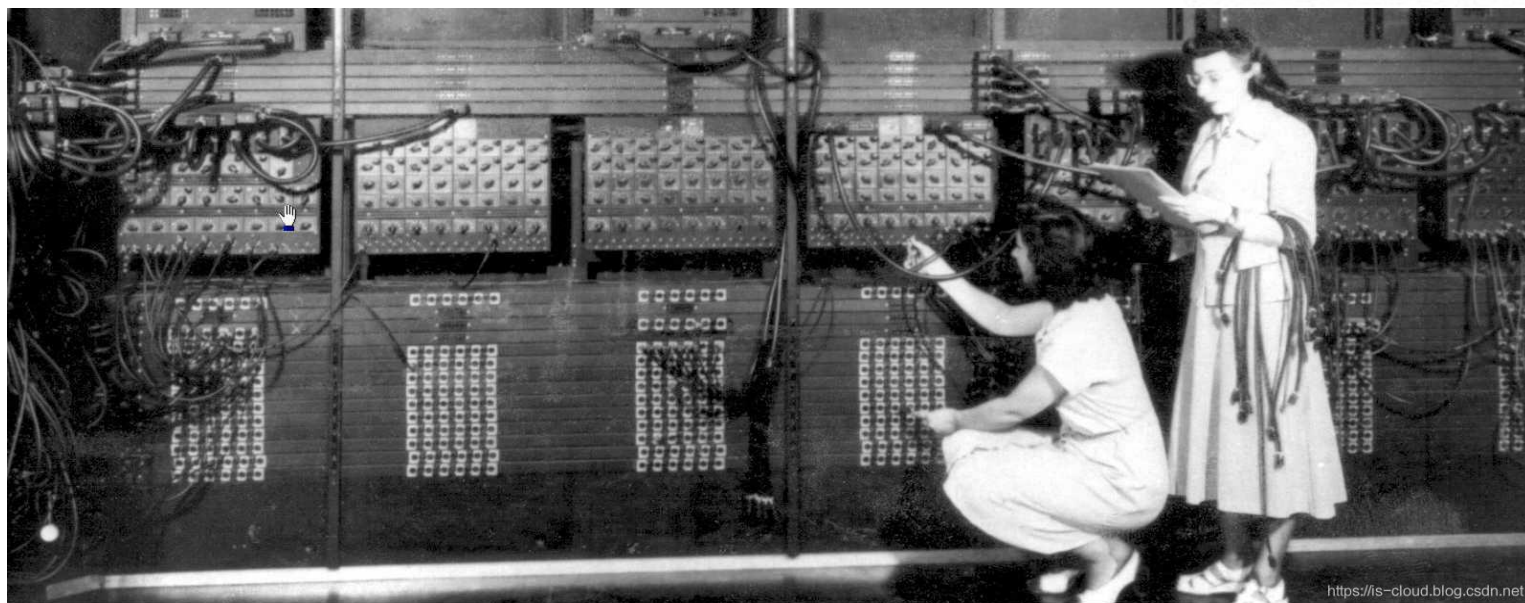
1.2.1 未配置OS的计算机系统

➤ 主要矛盾

- 计算机处理能力的提高，手工操作的低效率
- 没有充分发挥CPU的运算能力
- 用户独占全机的所有资源

➤ 提高效率的途径

- 专门的操作员，批处理



1.2.1 未配置OS的计算机系统

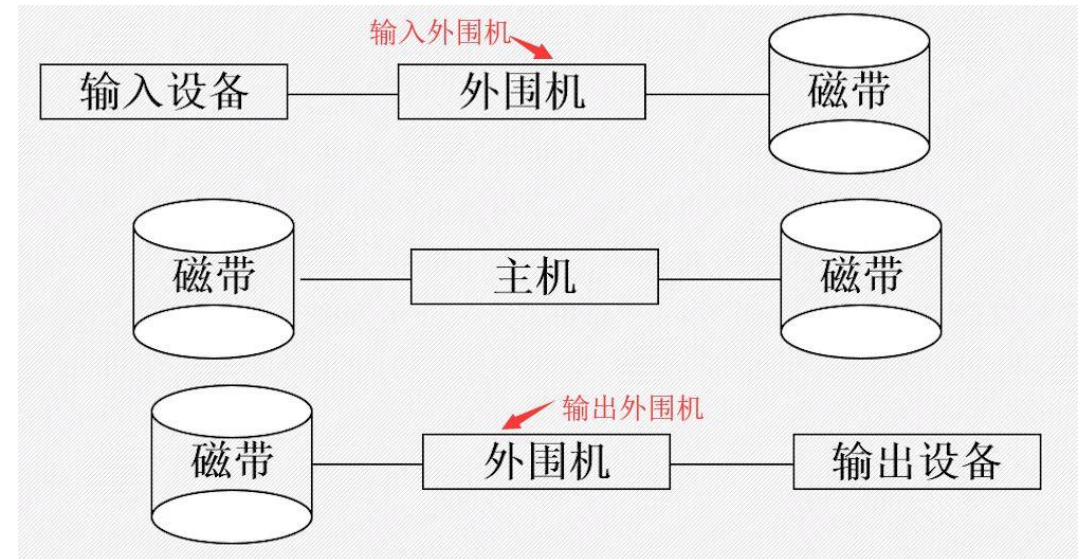
◆20世纪50年代末（电子管），集中计算（计算中心）

➤ 脱机输入输出方式（off-line I/O）

- 事先将装有用户程序和数据的数据带装入纸带输入机
- 外围机：把纸带/卡片上的数据/程序输入到磁带上
- CPU从磁带上高速地将数据调入内存
- 输出：内存→磁带→输出设备

➤ 优点

- 减少CPU的空闲时间
- 提高I/O速度



1.2.2 单道批处理系统

◆ 单道批处理系统：simple batch processing, uniprogramming

- 50年代末 ~ 60年代中 (**第二代晶体管计算机**)：50年代晶体管发明，计算机比较可靠，可成批地生产，用户可指望计算机长时间运行，完成一些工作
- 设计人员、生产人员、操作人员、程序人员和维护人员之间第一次有了明确的**分工**
- 利用**磁带**把若干个作业分类编成**作业执行序列**，每个批作业由一个专门的**监督程序** (Monitor) 自动依次处理
- 可使用**汇编语言**和**高级语言**开发

1.2.2 单道批处理系统

◆ 联机批处理

- 用户**提交**作业：以**纸带**或**卡片**为介质
- 操作员**合成批作业**：结果为**磁带**介质
- 作业**构成**：**程序**、程序使用的**数据**以及**作业控制卡**或作业**说明书**
- 用户使用系统提供的**作业控制语言命令**通过作业控制卡或作业说明书的形式把对作业的控制意图提交给**监控程序**
- **批作业处理**：对批作业中的每个作业进行相同的处理，即从磁带读入用户作业和编译链接程序，编译链接用户作业，生成可执行程序；启动执行；执行结果输出
- **问题**：慢速输入输出处理仍直接由主机来完成，输入输出时，CPU处于等待状态

1.2.2 单道批处理系统

◆ 脱机批处理

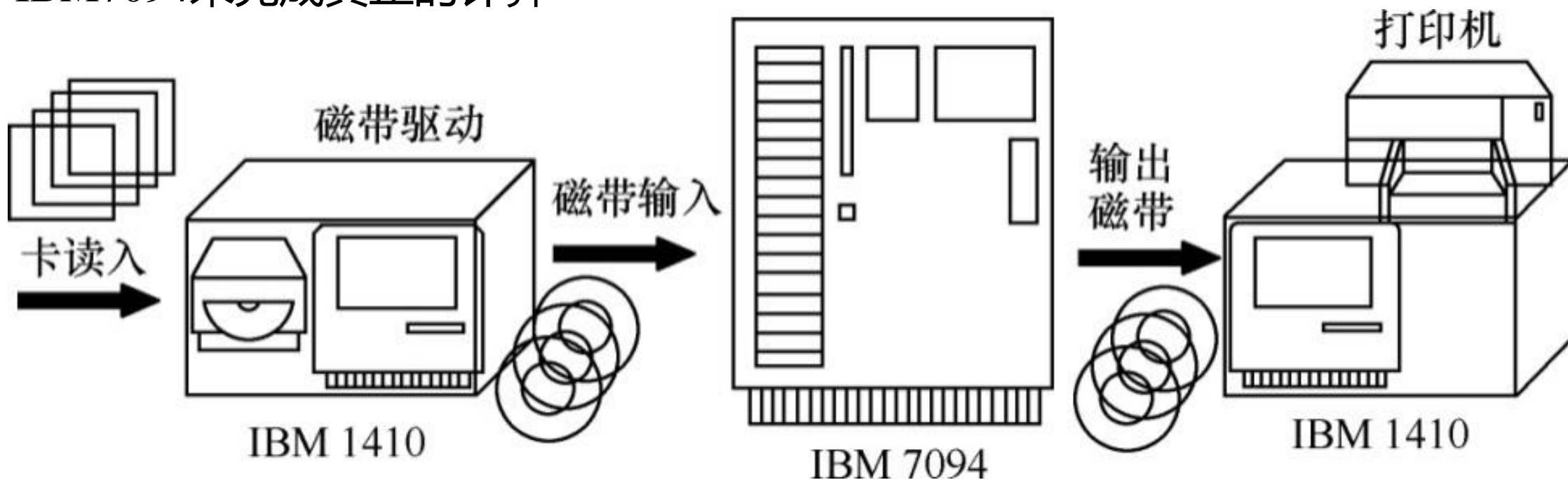
- 利用**卫星机**完成输入输出功能，主机完成计算，主机与卫星机可**并行工作**
- **卫星机**：完成面向用户的输入输出（纸带或卡片），中间结果暂存在**磁带或磁盘**上。
- 作业控制命令由**监督程序**(monitor)来执行，完成如装入程序、编译、运行等操作。
- **优点**：同一批内各作业的自动依次更替，改善了主机CPU和I/O设备的使用效率，**提高了吞吐量**。
- **缺点**：磁带或磁盘需要**人工装卸**，作业需要**人工分类**，监督程序易遭到用户程序的**破坏**（由人工干预才可恢复）。

1.2.2 单道批处理系统

◆ 脱机批处理

➤ 脱机输入/输出实例

IBM 7094机引入了I/O 处理机概念，其在输入时收集全部作业，用一台相对便宜的计算机（如IBM 1401机）将它们读到磁带上，另外用较昂贵的计算机，如IBM7094来完成真正的计算



1.2.2 单道批处理系统

◆ 单道批处理的特点

- **优点：**系统吞吐量大，资源利用率高
- **缺点：**作业的周转时间长，用户无法实现对作业的控制
- **适合**处理计算量大的、自动化程度高的成熟的作业
- CPU和I/O设备使用**忙闲不均**（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

1.2.3 多道批处理系统

◆多道批处理(Multiprogrammed Batch Processing System)

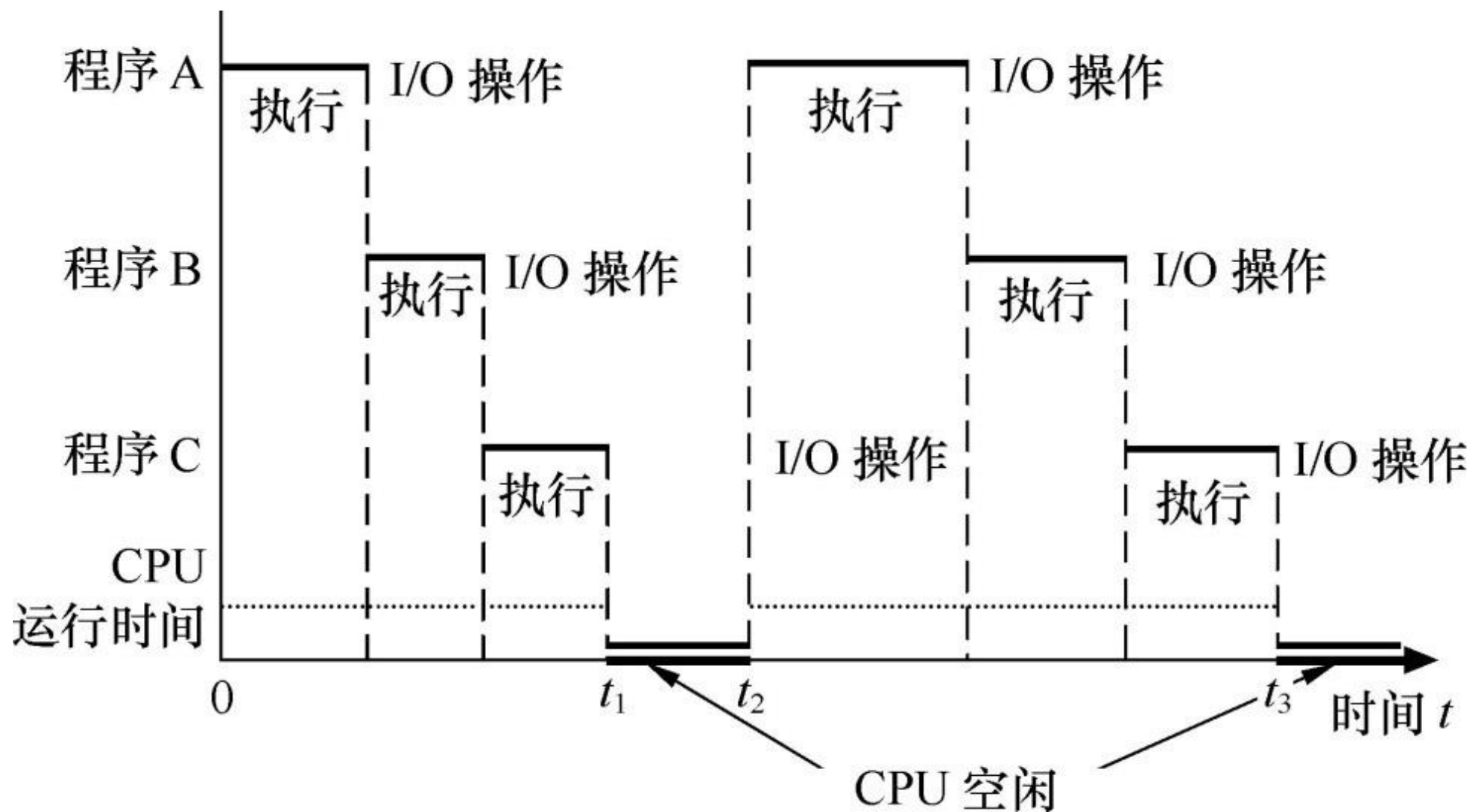
- 在内存中同时存放若干个作业，**共享系统资源并同时运行**
- 在该系统中，用户所提交的作业都先存放在外存上并排成一个队列，称为**“后备队列”**
- 由**作业调度程序**按一定的算法从后备队列中选择若干个作业调入内存，使它们共享CPU和系统中的各种资源。

◆多道批处理的运行特征

- **多道**：内存中同时存放几个作业
- 宏观上**并行运行**：都处于运行状态，但都未运行完
- 微观上**串行运行**：各作业交替使用CPU

1.2.3 多道批处理系统

◆ 多道批处理系统示意图



1.2.3 多道批处理系统

◆ 多道批处理系统的**特征**

- 多道性 无序性 调度性

◆ 优点

- **资源利用率高**：CPU和内存利用率较高
- **系统吞吐量大**：系统吞吐量是指系统在单位时间内所完成的总工作量（作业的数量）

◆ 缺点

- 用户**交互性差**：整个作业完成后或中间出错时，才与用户交互，不利于调试和修改
- 作业平均**周转时间长**：短作业的周转时间显著增长

1.2 多道程序设计例

◆ 若主存中有3道程序A、B、C，它们按A、B、C优先次序运行，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？

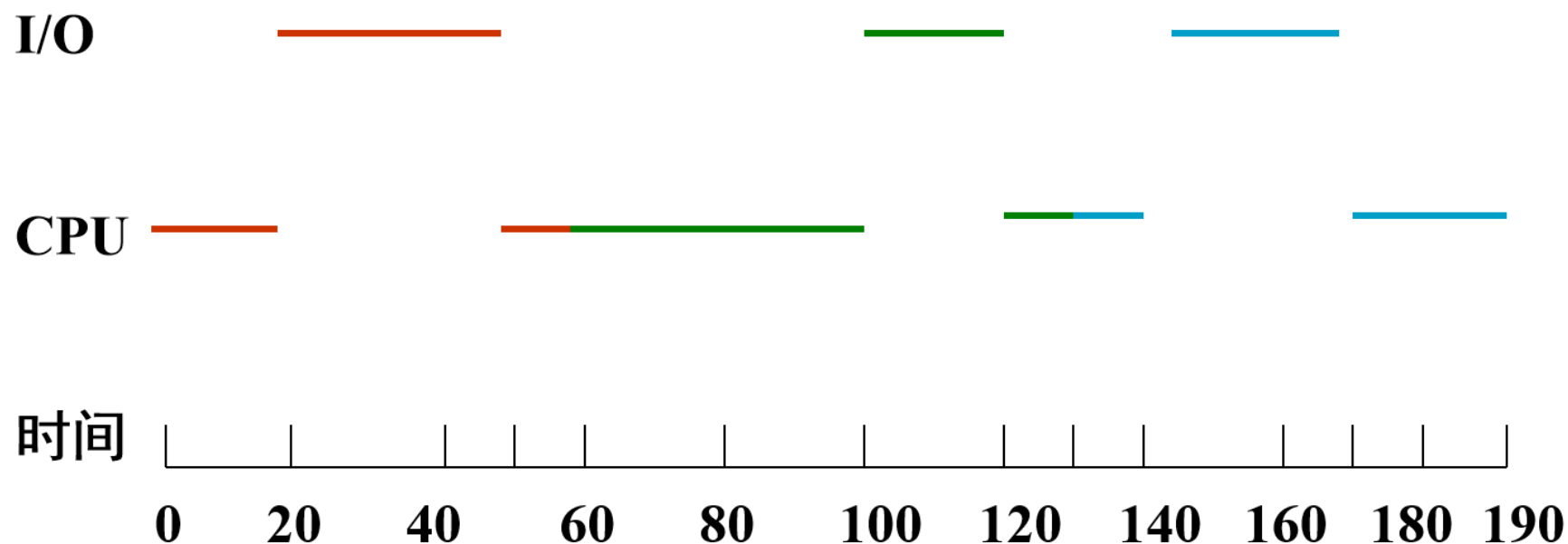
1.2 多道程序设计例

◆ 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)

◆ 单道CPU利用率为 $(190 - 80) \div 190 = 57.9\%$



单道运行时间关系图

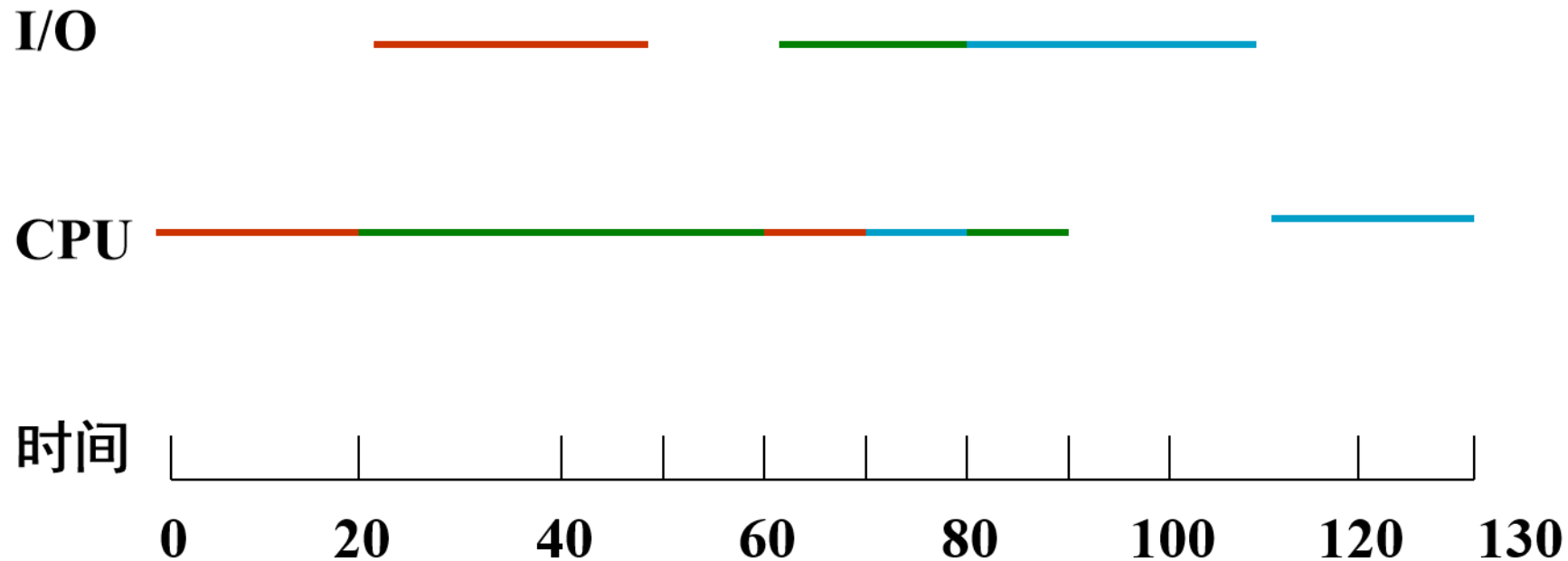
1.2 多道程序设计例

◆ 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)

◆ 多道CPU利用率为 $(130 - 20) \div 130 = 84.6\%$



多道运行时间关系图（不抢占）

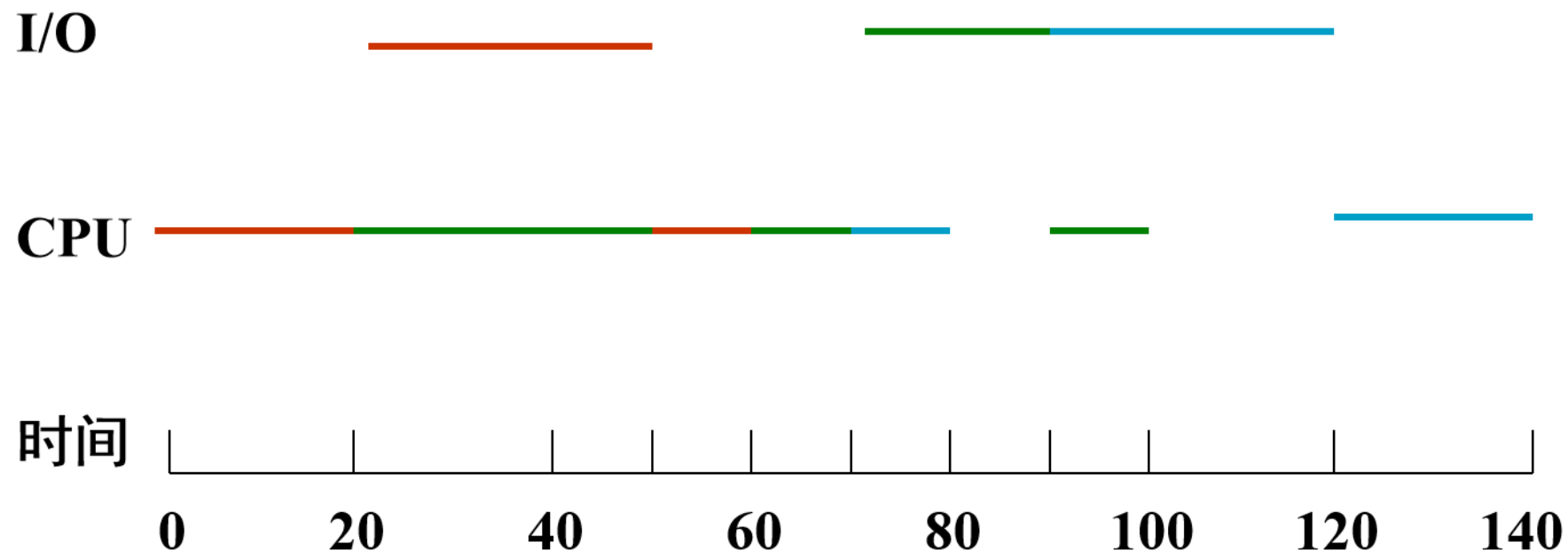
1.2 多道程序设计例

◆ 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)

◆ 多道CPU利用率为 $(140 - 30) \div 140 = 78.6\%$



多道运行时间关系图 (抢占)

1.2.4 分时系统 Time-Sharing System

- 推动**多道批处理系统**发展的主要动力：提高资源利用率和系统吞吐量 **(系统需求)**
- 推动**分时系统**形成和发展的主要动力，则是**用户的需求**
- 用户的**需求**具体表现在以下几个方面：
 - 人机交互
 - 共享主机
 - 便于用户上机
- 分时系统实现中的**关键问题**：
 - **及时接收**：及时接收用户的命令或数据，解决的办法是配置**多路卡**
 - **及时处理**：及时处理用户命令，应该使所有的用户的作业都直接进入**内存**；在很短的时间内使每个作业都运行一次

1.2.4 分时系统 Time-Sharing System

- ◆ “分时”指多个用户分享使用同一台计算机，也指多个程序分时共享硬件和软件资源
 - **多个用户分时**：多个应用程序同时在内存在，分别服务于不同用户，有用户输入时由CPU执行，处理完一次用户输入后程序暂停，等待下一次用户输入——时走时停
 - **前台和后台程序分时**：后台程序不占用终端输入输出，不与用户交互；现在的图形用户界面(GUI)，除当前交互的程序(输入焦点)之外，其他程序均作为后台
 - **时间片(time slice)分配**：各个程序在CPU上执行的轮换时间
 - **抢先式 preemptive**：OS强迫出让CPU；
 - **非抢先式 non-preemptive**：程序主动出让CPU；

1.2.4 分时系统 Time-Sharing System

- **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。宏观上是多个用户同时工作，共享系统资源，而微观上则是每个用户作业轮流运行一个时间片。多路性即同时性，它提高了资源利用率，从而促进了计算机更广泛的应用。
- **独立性：**每个用户各占一个终端，彼此独立操作，互不干扰。因此用户会感觉到就象他一人独占主机。
- **交互性：**用户可通过终端与系统进行广泛的人机对话。用户可以请求系统提供多方面服务，如文件编辑，数据处理和资源共享等。
- **及时性：**用户的请求能在很短时间内获得响应，此时间间隔是以人们所能接受的等待时间来确定的，通常为1-2秒。

1.2.4 分时系统 Time-Sharing System

- **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。宏观上是多个用户同时工作，共享系统资源，而微观上则是每个用户作业轮流运行一个时间片。多路性即同时性，它提高了资源利用率，从而促进了计算机更广泛的应用。
- **独立性：**每个用户各占一个终端，彼此独立操作，互不干扰。因此用户会感觉到就象他一人独占主机。
- **交互性：**用户可通过终端与系统进行广泛的人机对话。用户可以请求系统提供多方面服务，如文件编辑，数据处理和资源共享等。
- **及时性：**用户的请求能在很短时间内获得响应，此时间间隔是以人们所能接受的等待时间来确定的，通常为1-2秒。

1.2.5 实时系统 Real-Time System

- **要求：**响应时间短，系统可靠性高
- **任务的类型（按任务执行时是否呈现周期性来划分）**
 - 周期性实时任务
 - 非周期性实时任务：截止时间(deadline)，开始截止时间（任务在某时间以前必须开始执行）和完成截止时间（任务在某时间以前必须完成）
- **任务的类型（根据对截止时间的要求来划分）**
 - 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；
 - 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 Real-Time System

- **快速的响应时间**：实时系统是为了提高系统响应时间而设计的OS，特别是实时控制系统，对外部事件的响应要十分及时迅速。外部事件往往以中断方式通知系统，系统有较强的中断处理能力，实时系统的设计也以“事件驱动”方式来设计。
- **有限的交互能力**：实时系统（如实时信息处理系统）一般是专用系统，它能提供人机交互方式，但用户只能访问系统中某些特定的专用服务程序，不能象分时系统一般向终端用户提供多方面服务。
- **高可靠性**：批处理系统和分时系统虽也要求系统可靠，相比之下，实时系统则要求系统高度可靠。因此实时系统中往都采用双机系统，多级容错措施来保证系统和数据的安全。

1.4.4 OS基本类型

◆ 实时系统与批处理系统和分时系统的区别

- **专用系统：**许多实时系统是专用系统，而批处理与分时系统通常是通用系统。
- **实时控制：**实时系统用于控制实时过程，要求对外部事件的迅速响应，具有较强的中断处理机构。
- **高可靠性：**实时系统用于控制重要过程，要求高度可靠，具有较高冗余。如双机系统。
- **事件驱动和队列驱动：**实时系统的工作方式是接受外部消息，分析消息，调用相应处理程序进行处理。
- **可与通用系统结合成通用实时系统：**实时处理前台作业，批处理为后台作业。

1.2.6 微机操作系统

◆ 微机操作系统(microcomputer operating system): 配置在微型机上的操作系统

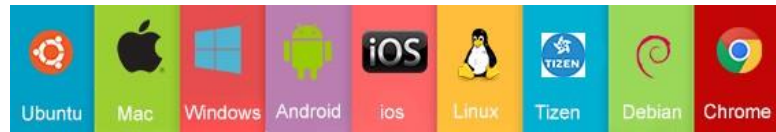
只允许一个用户上机，且只允许用户程序作为一个任务运行
代表：CP/M、MS-DOC。

只允许一个用户上机，但允许用户把程序分为若干个任务，使它们并发执行。
代表：Windows

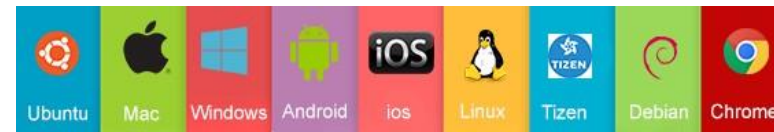
允许多个用户通过各自的终端，使用同一台机器，共享主机系统中的各种资源，而每个用户程序又可进一步分为几个任务，使它们能并发执行。
代表：Unix OS



操作系统引论



1.3 OS的基本特性



并发性：进程



两个或两个以上的事件在同一时间段内发生

共享性：资源



计算机系统资源能够被并发执行的多个进程共同使用

虚拟性：存储



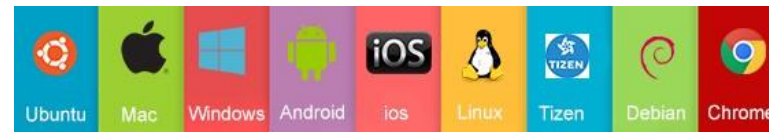
系统通过某种技术将一个实际存在的实体变成多个逻辑上的对应体

异步性：进程



也称为随机性，是指多道程序环境中多个进程的执行、推进和完成时间都是随机、交替、不可预测的

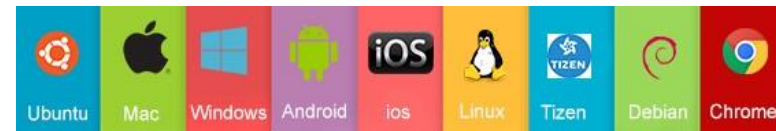
1.3 OS的基本特性



◆ 并发性：进程

- **意义：**共享是指系统中的所有资源**不再为**一个程序所**独占**，而是供同时存在于系统中的**多道程序**所**共同使用**。根据资源属性不同，可有互斥共享和同时共享两种不同的共享方式。
- **互斥共享：**在一段时间内，**只允许一个进程访问该资源**。统中应建立一种机制以保证多个进程对这类资源的互斥访问
- **同时共享：**允许在一段时间内由多个进程“**同时**”进行访问。这里所谓的“同时”，在单处理机环境下是宏观意义上的，而在微观上，这些进程对该资源的访问是**交替**进行的。典型的可供多个进程“同时”访问的资源是磁盘设备。一些用重入码编写的文件也可以被“同时”共享，即允许若干个用户同时访问该文件

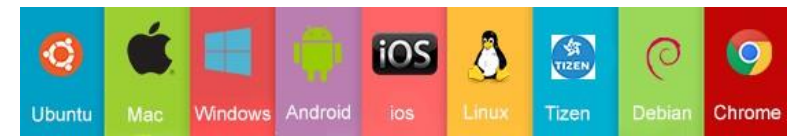
1.3 OS的基本特性



◆ 共享性：资源

- **意义**：系统中程序能并发执行的特征，使得OS能有效**提高**系统资源**利用率**，**增加系统吞吐量**
- **引入进程**：引入之前，属于同一个应用程序的计算程序和I/O程序之间只能是**顺序执行**，即只有在计算程序执行告一段落后，才允许I/O程序执行；反之，在程序执行I/O操作时，计算程序也不能执行。**引入进程**后，可以为计算程序和I/O程序分别建立一个进程(Process)后，这两个进程便可**并发执行**
- **并行性**是指两个或多个事件在**同一时刻发生**
- **并发性**是指两个或多个在同一时间间隔内发生
- 在多道程序环境下，**并发性**是指宏观上在**一段时间内**有多道程序在**同时执行**。但在单处理机系统中，每一个时刻仅能执行一道程序，故微观上，这些程序是在**交互执行**

1.3 OS的基本特性



$$\lim_{x \rightarrow 0} \frac{\sqrt[4]{1+x} - 1}{x} = \lim_{x \rightarrow 0} \frac{(1+x)^{\frac{1}{4}} - 1}{x} = \frac{1}{4}$$

$$\sqrt[3]{\frac{12-2x}{x-1}} + \sqrt[3]{\frac{x-1}{12-2x}} = \frac{5}{2}, \text{ if } \sqrt[3]{\frac{12-2x}{x-1}},$$

$$\text{then } u + \frac{1}{u} = \frac{5}{2} \Rightarrow 2u^2 - 5u + 2 = 0 \Rightarrow$$

$$u_1 = 2, u_2 = \frac{1}{2} \Rightarrow \sqrt[3]{\frac{12-2x}{x-1}} = 2,$$

$$\frac{12-2x}{x-1} = 8 \Rightarrow x_1 = 2,$$

$$D = (-2)^2 - 4 \cdot 5 \cdot (-39) = 4 + 20 \cdot 39 = 784,$$

$$x_1 = \frac{2+28}{2 \cdot 5} = \frac{30}{10} = 3$$

$M(3, 7, 1)$
 $d = \sqrt{(3 - \frac{33}{13})^2 + (7 - \frac{83}{13})^2 + (1 - \frac{37}{13})^2}$
 $N(\frac{33}{13}, \frac{83}{13}, \frac{37}{13})$
 6^2

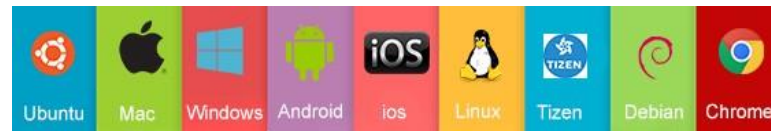
$f(x) = x^3 - 3x$

$$\int (5x^3 + 4x^2 - 7x + 12) dx = \frac{5}{4}x^4 + \frac{4}{3}x^3 - \frac{7}{2}x^2 +$$

$$y = x^4 - 14x^3 + 60x^2 + 8x - 9 \quad \int dx$$

物理实体(前者)是实的 即实际存在的;

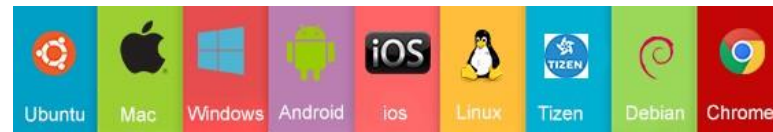
1.3 OS的基本特性



◆虚拟性：存储

- **意义：**指通过某种技术把一个**物理实体**变成若是个**逻辑**上的对应物。物理实体是实际存在的；**逻辑**物体是用户感觉到的，是**虚拟**的。例如在单CPU多道分时系统中，通过多道程序技术和分时技术可以把一个物理CPU虚拟为多台逻辑上的CPU，使每个终端用户都认为有一台“独立”的CPU为它运行，用户感觉的CPU是虚拟CPU。
- **时分复用技术：**多道程序设计和一台处理机，分时为多个用户使用，从而实现一台物理处理机被虚拟成多个逻辑处理机。
- **空分复用技术：**利用存储器的空闲空间分区域存放和运行其他的多道程序，以此提高内存的利用率。

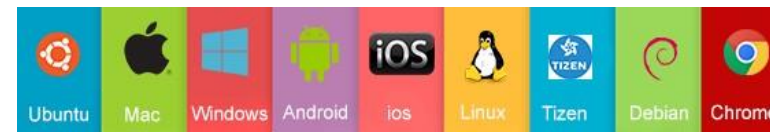
1.3 OS的基本特性



◆异步性：进程

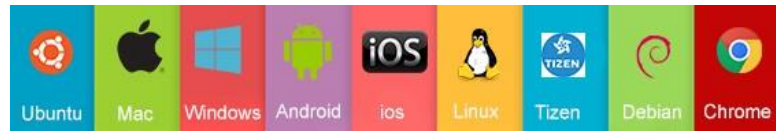
- **意义：**在多道程序环境下，允许多个程序**并发执行**，但由于资源等因素的限制，程序的执行并非“一气呵成”，而是以“**走走停停**”的方式运行，即程序是以**异步方式**运行的。
- 在**单处理机**环境下，由于系统中只有一台处理机，因而每次只允许一个进程执行，**其余进程只能等待**。
- 当正在执行的进程提出某种资源要求时，如打印请求，而此时打印机正在为其它进程打印，由于打印机属于**临界资源**，因此正在执行的进程必须等待，并释放出处理机，直到打印机空闲，并再次获得处理机时，该进程方能继续执行。

1.3 OS的基本特性

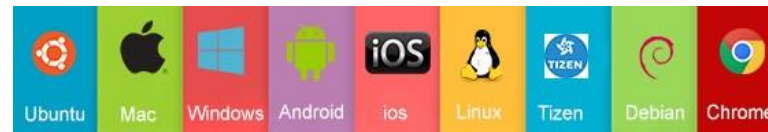


- **并发和共享关系**：并发和共享是操作系统的两个最基本的特性，它们又是互为存在条件。一方面资源共享是以程序（进程）的并发性执行为条件的，若系统不允许程序并发执行，自然不存在资源共享问题。另一方面若系统不能对资源共享实施有效管理，则也必将影响到程序并发执行。
- **不确定性（nondeterministic）**：每个程序（进程）执行速度和时间不确定，各程序（进程）之间推进序列也不确定。即是不可预测的；每个程序（进程）执行结果不确定，即对同一程序，给定相同的初始条件、在相同的环境下进行多次执行，却可能获得完全不同的结果，这也称为程序并发执行的不可再现性。执行结果的不确定性是绝对不允许的，这是操作系统为实现程序并发执行必须解决的问题。

操作系统引论



1.4 OS的主要功能

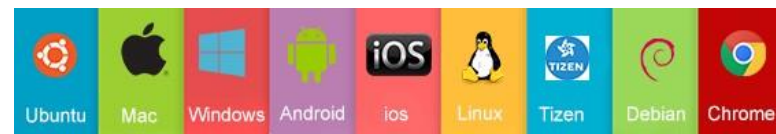


◆引入OS的目的：管理资源、提供接口

- 有条不紊地、高效地运行，提高系统资源的利用率和吞吐量
- 方便用户的使用

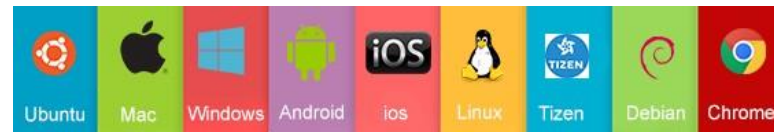


1.4.1处理器管理功能



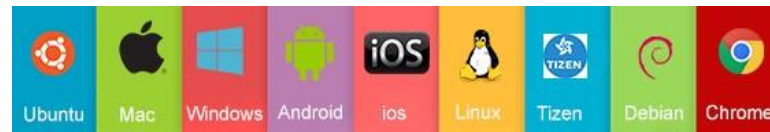
- 对处理机进行分配，并对其运行进行**有效的控制和管理**。在多道程序环境下，处理机的分配和运行以**进程为基本单位**，因而对处理机管理可归纳为对**进程的管理**
- **进程控制**：为作业**创建/撤销**进程，控制进程在运行过程中的**状态转换**，为进程创建**线程**以提高系统并发性
- **进程同步**：协调多个进程/线程的运行
 - **进程互斥**：互斥访问临界资源（**锁**）
 - **进程同步**：相互合作完成共同任务，由同步机构对执行顺序加以协调（**信号量机制**）
- **进程通信**：实现相互合作进程之间的**信息交换**
- **调度**：
 - **作业调度**：从后备队列中选择若干作业分配资源，调入内存，建立进程，插入就绪队列；
 - **进程调度**：从进程的就绪队列选一个进程为其分配处理器和运行环境

1.4.2 存储器管理功能



- 为多道程序的运行提供良好的环境，方便用户使用存储器，提高**存储器的利用率**，以及能从逻辑上来**扩充内存**
- **内存分配**：分配内存；提高存储器利用率；运行正在运行的程序申请附加的内存空间
 - **静态分配方式**：作业装入时确定内存空间大小，不允许附加新空间
 - **动态分配方式**：运行过程可申请附加空间，允许作业在内存移动
- **内存保护**：内存空间内运行；绝不允许用户程序访问OS的程序和数据；不允许用户程序转移到非共享的其他程序中执行
- **地址映射**：在硬件支持下，程序的逻辑地址→内存物理地址
- **内存扩充**：借助虚拟存储技术，从逻辑上扩充内存容量。
 - **请求调入功能**：可启动仅装入部分程序 and 数据的进程运行
 - **置换功能**：可将内存中一部分暂时不用的程序和数据调至硬盘

1.4.3 设备管理功能



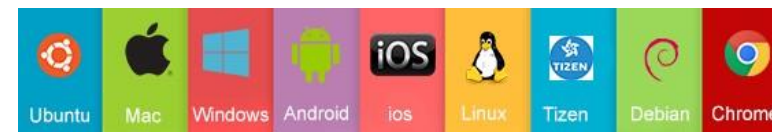
- 完成用户进程提出的**I/O请求**，为其分配所需I/O设备，完成指定I/O操作；**提高CPU和I/O设备的利用率和I/O速度**
- **设备控制方式**：控制方式；中断方式；直接存储器访问（DMA）方式；通道方式
- **缓冲管理**：在CPU和I/O设备之间引入缓冲，有效缓和两者速度不匹配的矛盾，提高**CPU利用率和系统吞吐量**。
 - **常见缓存机制**：单缓冲区、双缓冲区、循环缓冲、缓冲池
- **设备分配**：根据用户进程的请求、系统现有资源情况及设备分配策略为之分配其所需设备
- **设备处理（设备驱动）**：实现CPU和设备控制器之间的通信
- **磁盘存储器管理**：磁盘存储空间的划分，磁盘存储空间的分配和回收

1.4.4 文件管理功能



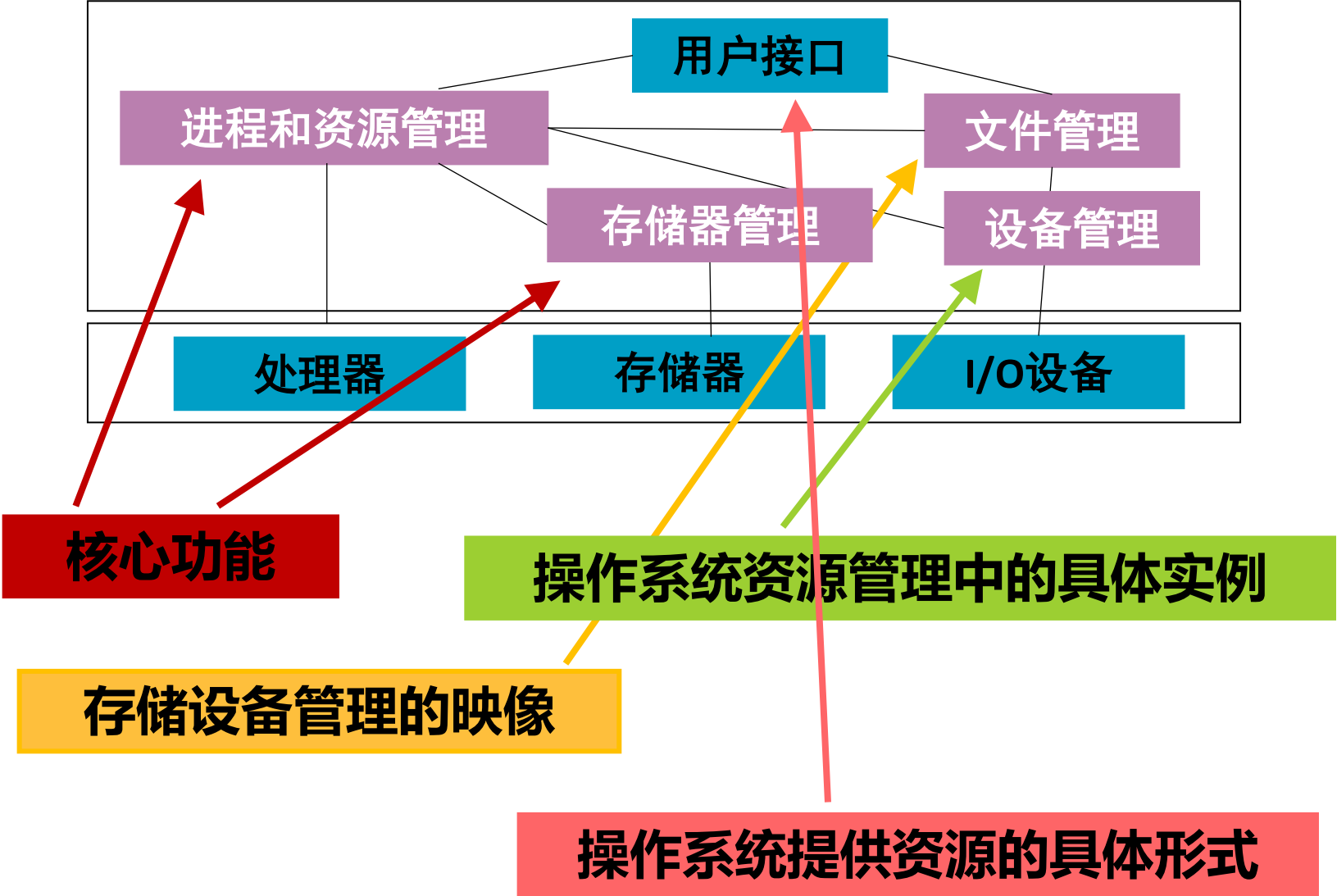
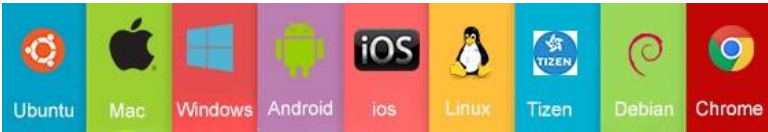
- 计算机系统的**软件信息**都以**文件形式**进行管理，操作系统中负责此任务的部分是**文件管理**，其任务是对**用户文件和系统文件**进行管理，以**方便用户使用**，并保证文件的**安全性**。
- **文件存储空间的管理**：为每个文件分配必要的外存空间，提高外存利用率；根据文件的组织方式合理地分配文件存储空间，有效地管理文件存储空间，实现用户访问文件的快速和有效性。
- **目录管理**：为每个文件建立目录项，实现文件的按名存取，为用户提供方便的文件系统接口，便于用户对文件进行操作。
- **文件的读/写管理**：根据用户请求，从外存中读取数据，或将数据写入外存。
- **文件保护**：提供文件的共享和保护，做到用户对文件的访问权限控制。

1.4.5 用户接口

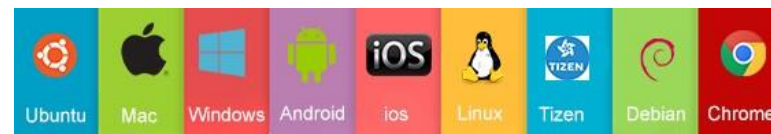


- 为了方便用户使用OS，OS向用户提供了“用户与OS的接口”
- **程序接口**：在用户编程应用上，操作系统为程序员提供系统资源**调用函数**，达到程序员在程序中方便使用系统资源的目的。
- **命令接口**：对系统的管理和应用，操作系统提供一套**系统命令**供系统管理员和用户使⽤。命令接口需要用户熟悉命令结构和命令形式。
- **图形接口**：对系统的管理和应用，操作系统以**图形窗口**方式提供给系统管理员和用户使⽤。图形接口操作简单、直观。

1.4.6 OS的主要功能



1.4.7 现代OS的新功能



➤ 系统安全

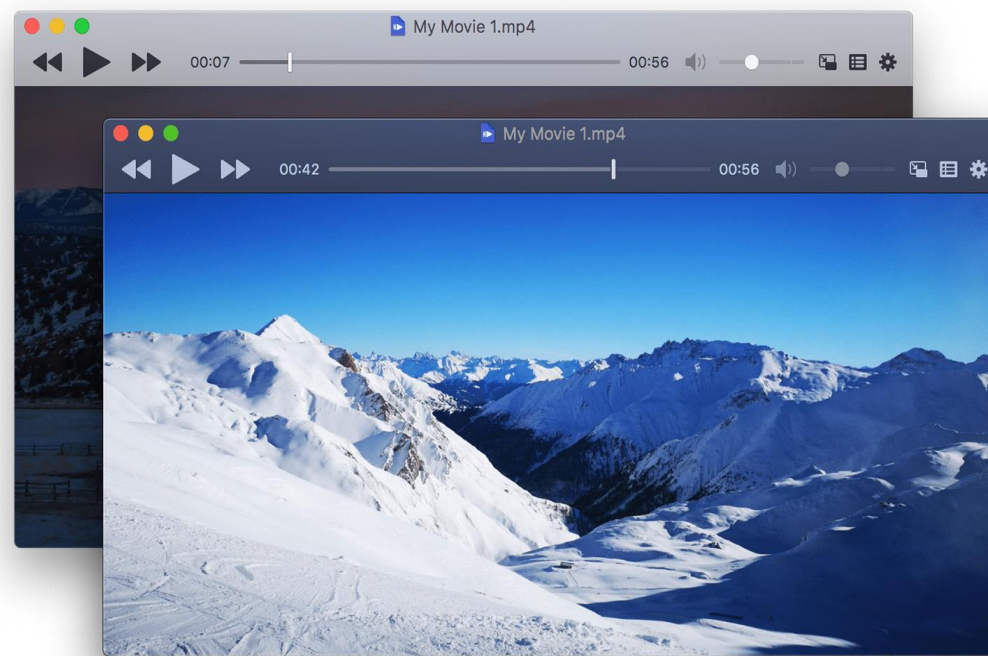
- 认证技术
- 密码技术
- 访问控制技术
- 反病毒技术

➤ 网络的功能和服务

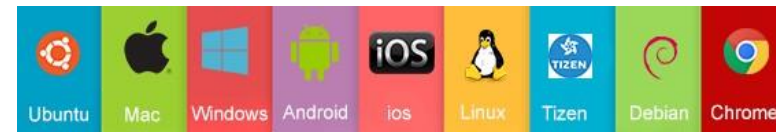
- 网络通信
- 资源管理
- 应用互操作

➤ 支持多媒体

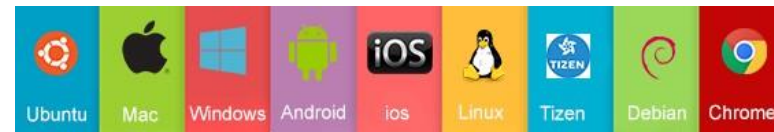
- 接纳控制功能
- 实时调度
- 多媒体文件的存储



操作系统引论



1.5.1 传统OS结构



- **无结构操作系统：** OS是为数众多的一组过程的集合，每个过程可以任意地相互调用其它过程，致使操作系统内部既复杂又混乱。
- **模块化结构OS：** 该技术基于“分解”和“模块化”原则来控制大型软件复杂度。为使OS具有较清晰的结构，OS不再是由众多的过程直接构成的，而是按其功能精心地划分为若干个具有一定独立性和大小的模块。关键问题：模块之间的接口；模块的独立性问题。
- **分层式结构OS：** 有序分层法，其思想与计算机网络体系结构非常相似。

1.5.2 C/S模式（客户端/服务器模式）

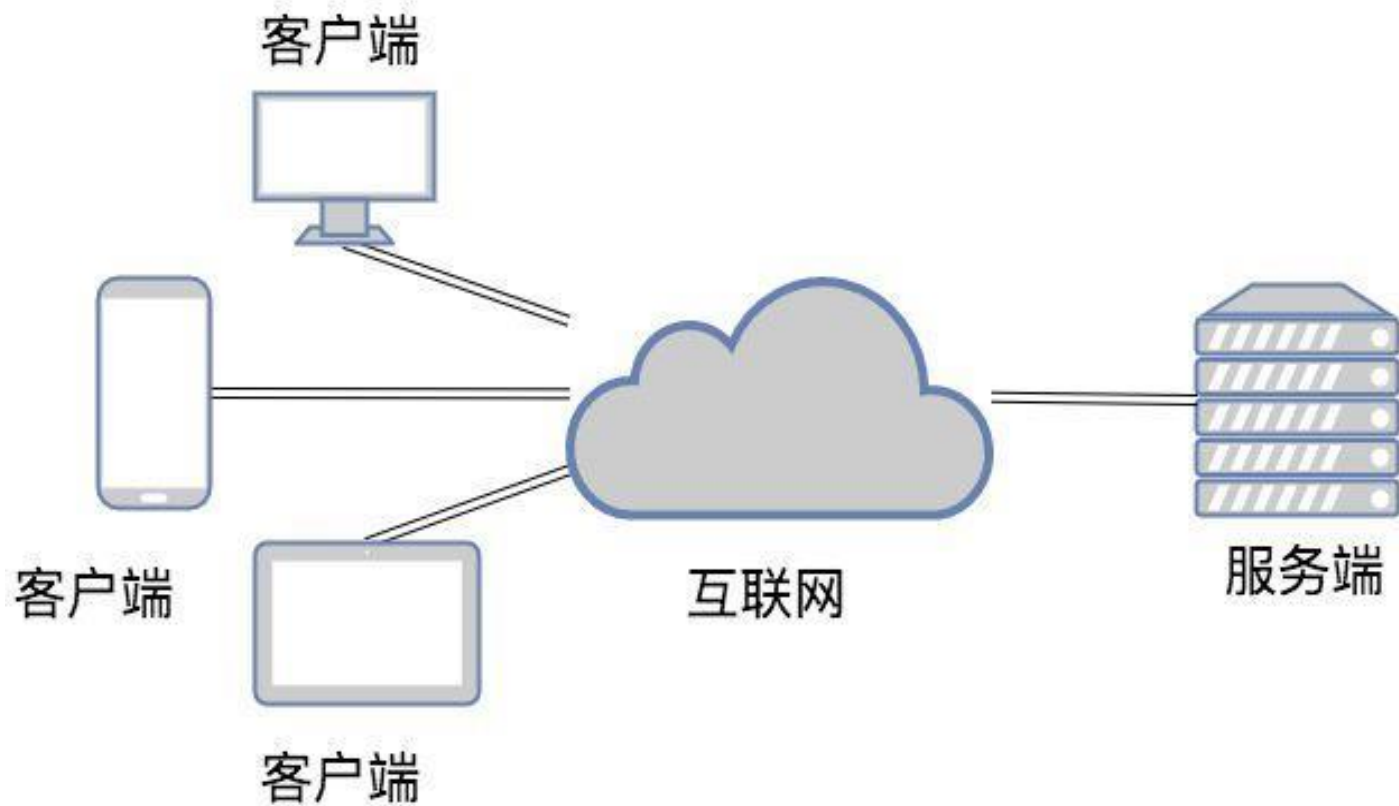
➤ **基本组成：** 客户机、服务器、网络系统

➤ **客户/服务器之间的交互：**

- 客户发送请求
- 服务器接收请求
- 服务器回送处理结果
- 客户机接收处理结果

➤ **优势分析：**

- 数据的分布处理和存储
- 便于集中管理
- 灵活性和可扩充性
- 易于改编应用软件

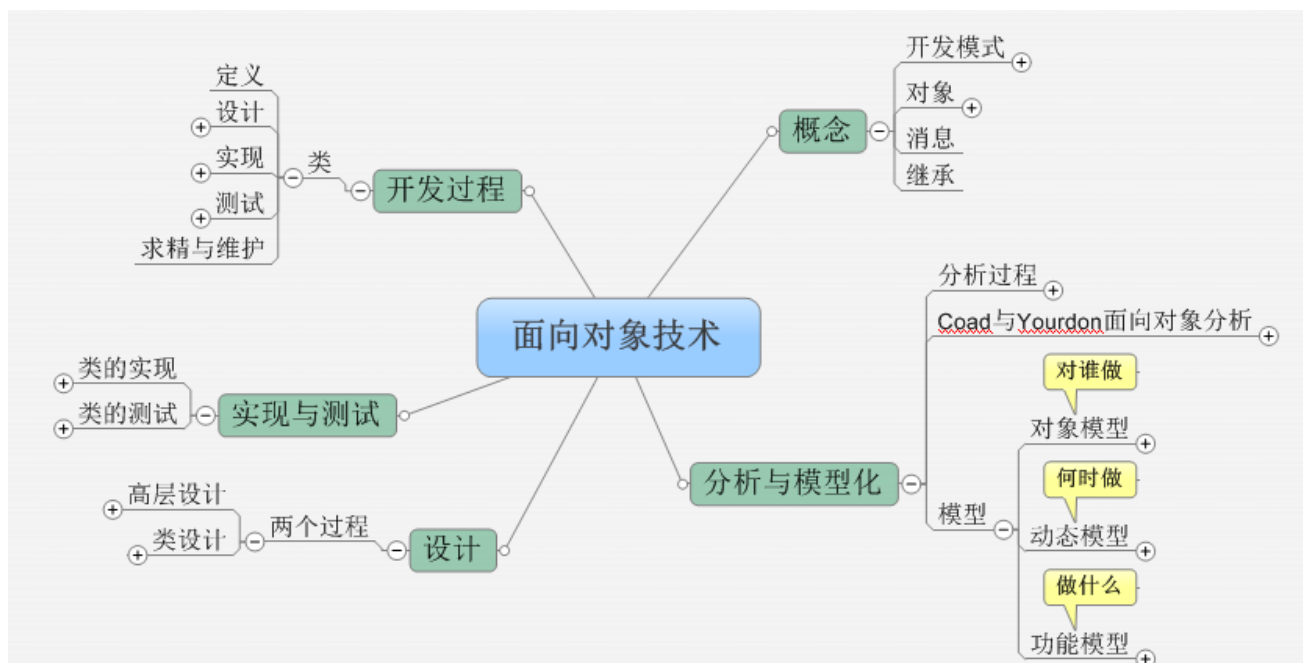


1.5.3 面向对象的程序设计

➤ **基本概念：**对象、对象类、继承

➤ **优势分析：**

- 通过“重用”提高产品质量和生产率
- 使系统具有更好的易修改性和易扩展性
- 更易于保证系统的“正确性”和“可靠性”



1.5.4 微内核OS结构

◆ 设计目的:

- 提高OS的正确性、灵活性、易维护性、可扩充性

◆ 基本概念

➤ 足够小的内核:

- 将操作系统中最基本的部分放入微内核, 如与硬件处理紧密相关的部分、一些较基本的功能、客户和服务端之间的通信

➤ 基于客户/服务器模式:

- OS绝大部分功能都放在微内核外面的一组服务器(进程)中实现, 运行在用户态
- 客户与服务端之间借助微内核提供的消息传递机制来实现信息交互

➤ 应用“机制与策略分离”原理

- 机制——实现某一功能具体执行机构; 策略——机制基础上实现功能优化
- 在微内核系统中, 机制放在微内核中, 内核足够小

➤ 面向对象技术

- “抽象”、“隐蔽”: 控制系统复杂性
- “对象”、“封装”、“继承”: 确保系统正确性、可靠性、易修改性、易扩展性

1.5.4 微内核OS结构

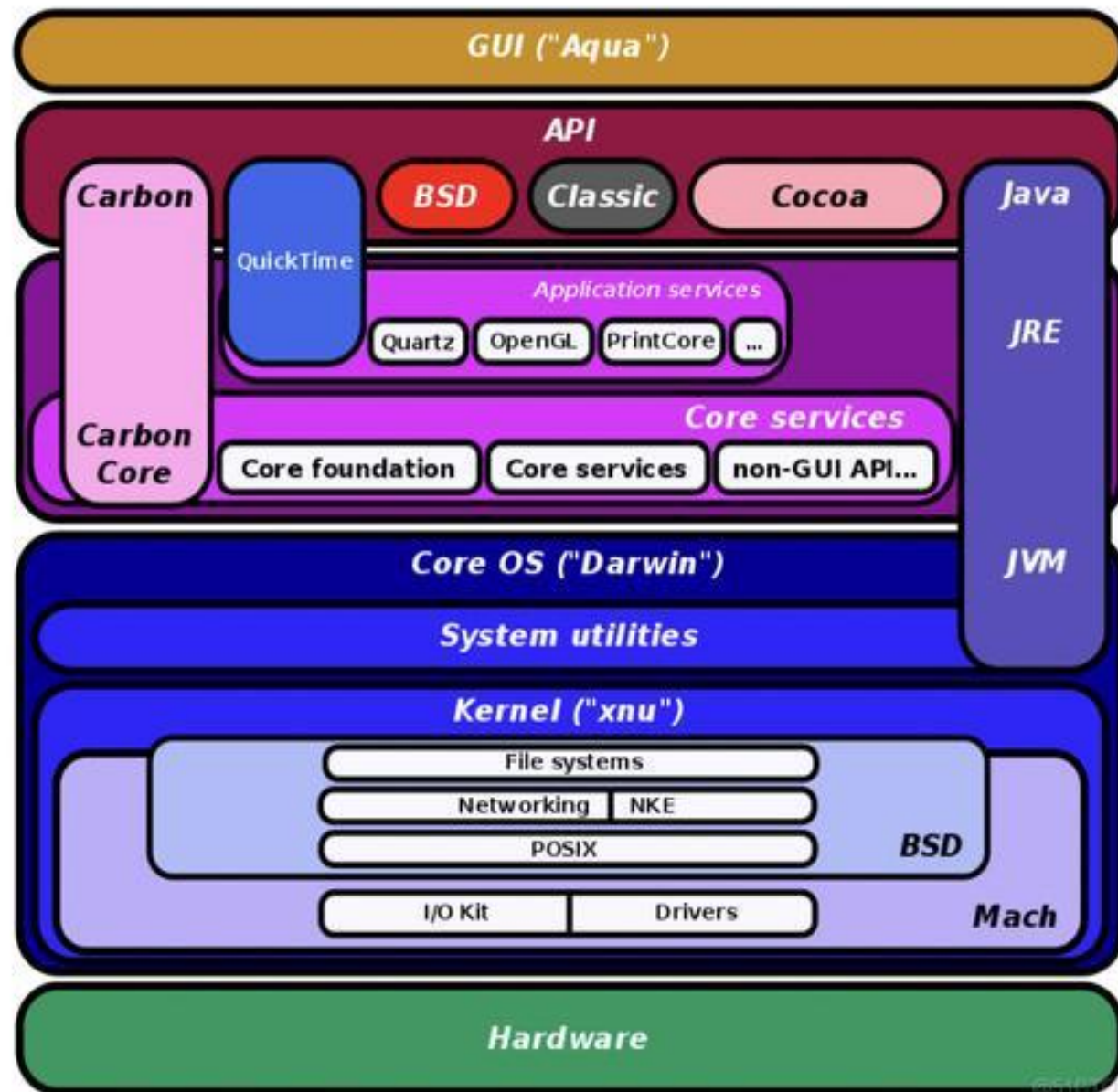
◆ 基本功能:

- 进程/线程管理
- 低级存储器管理
- 中断和陷入处理

◆ 优势分析:

- 提高了系统的可扩展性
- 增强了系统的可靠性
- 可移植性强
- 提供了对分布式系统的支持
- 融入了面向对象技术

◆ 存在问题分析: 运行效率降低



THEORY

PRACTICAL



(97年高级程序员试题2) 在有一台处理机CPU和两台输入输出设备IO1和IO2, 且能够实现抢先式多任务并行工作的多道程序环境内, 投入运行优行级由高到低的P1、P2、P3 三个作业。它们使用设备的先后顺序和占用设备时间分别是:

- 作业P₁: IO2(30ms)、CPU(10ms)、IO1(30ms)、CPU(10ms)
- 作业P₂: IO1(20ms)、CPU(20ms)、IO2(40ms)
- 作业P₃: CPU(30ms)、IO1(20ms)

在控制程序介入时间可以忽略不计的假设下, 作业P₁、P₂、P₃ 从投入到完成所用的时间分别是多少? 三个作业从投入运行到全部完成, CPU、IO1和IO2的利用率分别是多少? (假设在系统中仅有这三个作业投入运行, 各设备的利用率指该设备的使用时间同作业进程组全部完成所占用最长时间的比率。)

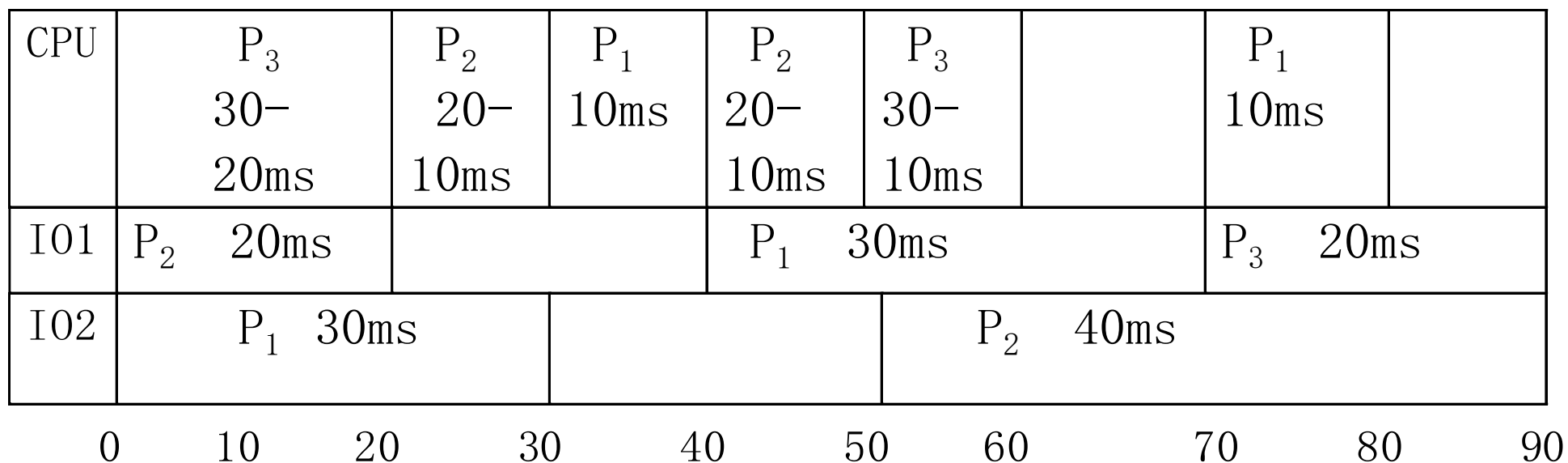
◆ 作业调度的方式：

- 输入输出设备非抢占方式调度；
- CPU有二种调度的方式：非抢占方式和抢占方式，由题义决定为抢先式。

◆ 选择调度作业原则：

- 优先权；
- 资源（CPU、I/O设备、内存等）满足要求；

◆ 调度时机：完成一阶段CPU计算或I/O



从图中可知：

P₁ 从投入到完成所用时间 $30 + 10 + 30 + 10 = 80\text{ms}$

P₂ 从投入到完成所用时间 $20 + 10 + 10 + 10 + 40 = 90\text{ms}$

P₃ 从投入到完成所用时间 $20 + 30 + 10 + 10 + 20 = 90\text{ms}$

CPU 利用率 $= 70 / 90 = 78\%$

I01 利用率 $= 70 / 90 = 78\%$

I01 利用率 $= 70 / 90 = 78\%$

1. 操作系统是一种.....A.....，在操作系统中采用多道程序设计方式能提高CPU和外部设备的.....B.....。一般来说，为了实现多道程序设计，计算机需要有.....C.....。

A: (1)通用软件；(2)系统软件；(3)应用软件；(4) 软件包。

B: (1)利用效率；(2)可靠性；(3)稳定性；(4)兼容性。

C: (1)更大的内存；(2)更快的外部设备；(3)更快的CPU；(4)更先进的终端；

2. 分时系统中，为使多个用户能够同时与系统交互，最关键的问题是.....A.....，当用户数目为100时，为保证响应不超过2秒；此时的时间片最大应为.....B.....。

A: (1)计算机具有足够的运行速度；(2)内存容量应足够大；(3)系统能及时地接收多个用户输入；(4)能在一短的时间内，使所有用户程序都能运行；(5)能快速进行内外存对换

B: (1)10ms；(2)20ms；(3)50ms；(4)100ms；(5)200ms

3.在设计分时操作系统时，首先要考虑的是.....A.....；在设计实时操作系统时，首先要考虑的是.....B.....；在设计批处理系统时，首先要考虑的是.....C.....。

(1)灵活性和可适应性；(2)交互性和响应时间；(3)周转时间和系统吞吐量；(4)实时性和可靠性。

4. 试从目标、多路性、独立性、交互性、及时性和可靠性多方面来比较批处理系统、分时系统及实时系统。通过比较，请写出这三种系统各适用于什么场合。

5.为了提高计算机的处理机和外部设备的利用率，把多个程序同时放入主存储器，在宏观上并行运行是.....A.....；把一个程序划分成若干个同时执行的程序模块的设计方法是.....B.....；多个用户在终端设备上的交互方式输入、排错和控制其程序的运行是.....C.....；由多个计算机组成的一个系统，这些计算机之间可以通信来交换信息，互相之间无主次之分，它们共享系统资源，程序由系统中的全部或部分计算机协同执行，管理上述计算机系统的操作系统是.....D.....；有一类操作系统的系统响应时间的重要性超过系统资源的利用率，它被广泛地应用于卫星控制、导弹发射、飞机飞行控制、飞机订票业务等领域是.....E.....。

A--E: ① 分时OS ② 实时OS ③ 批处理系统 ④ 网络OS ⑤ 分布式OS ⑥ 单用户OS ⑦ 多重程序设计 ⑧ 多道程序设计 ⑨ 并发程序设计