



河海大学

计算机与信息学院

操作系统 (0601009)

河海大学“课程思政”示范课程

授课教师:	张鹏程、陆佳民
Email:	pchzhang@hhu.edu.cn
QQ:	185319755、1284762490
办公室:	勤学楼4515、4121



回顾

- 操作系统的基础抽象包括哪三种？
- 操作系统的其它抽象？举例
- 操作系统定义？
- 四种常用观点来看待操作系统的作用？
- 操作系统的五大功能？
- 操作系统的三大特性？
- 什么是并发性？和并行性的区别？
- 什么是共享性？
- 什么是异步性？



1.2 操作系统形成与发展

1.2.1 人工操作阶段

1.2.2 执行系统阶段

1.2.3 多道程序设计与操作系统形成

1.2.4 操作系统发展与分类



1.2 操作系统形成与发展

- 辩证唯物主义事物的发展规律：对立统一规律是唯物辩证法的根本规律，它揭示出自然界、人类社会和人类思维等领域的任何事物都包含着内在的矛盾性，对立统一规律揭示了普遍联系的根本内容和事物发展的内在动力，揭示了事物发展的动力和源泉，揭示了发展和联系的本质
- 操作系统的发展历程符合辩证唯物主义事物的发展规律
 - 机器和人之间的矛盾
 - 软件和硬件之间的矛盾

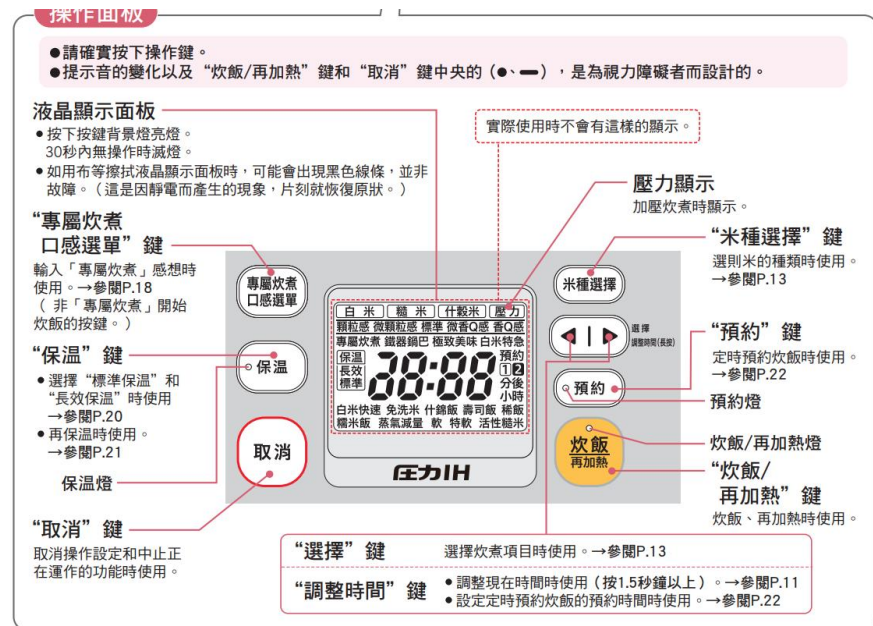


1.2.1 人工操作阶段

- 任何机器都有操作系统，

如电饭锅：

- 烹饪类型、预约和保温时间等
- 启动按钮
- 通过指示灯来显示工作状态

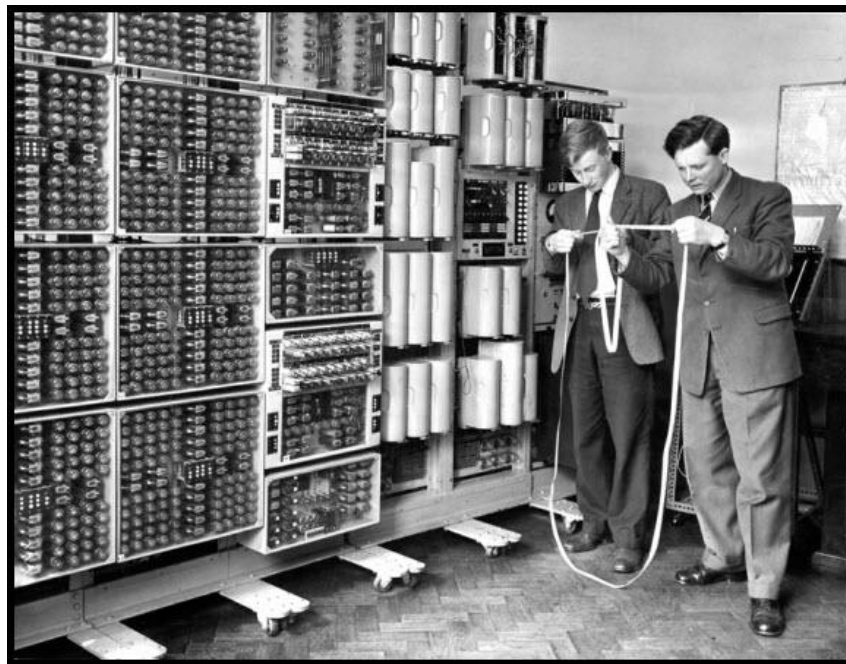


开关表示，按钮控制，亮灯显示



1.2.1 人工操作阶段

- 第一代计算机：从计算机诞生到50年代中期的计算机
- 没有显式的，自动化的操作系统



开关表示，按钮控制，亮灯显示



1.2.1 人工操作阶段

- 示例视频：课堂在线 -> 资料 -> 课程公共资源
 - “Altair 8800 - 演示”：展示1974年第一台PC Altair 8800的操作方式
 - bilibili: av5150794
 - “NO_OS 的程序执行过程”：展示利用1959年的IBM 1401，来执行矩阵计算的过程
 - 视频来源：[The IBM 1401 compiles and runs FORTRAN II](#)
 - bilibili: av24487822



1.2.1 人工操作阶段

- 作业和程序（进程）的区别
 - 一个进程是一个程序对某个数据集的执行过程。
 - 一个作业是用户需要计算机完成的某项任务，是要求计算机所做工作的集合。一个作业中至少包含一个程序。

一个作业的执行过程（样例）：

装入 load - 编译 compile - 连接 link - 执行 execute - 输出 output



1.2.1 人工操作阶段

- 人工控制和使用计算机的作业过程大致如下：

1. **coding**: 人工把源程序用穿孔机穿制在卡片或纸带上；
2. **load**: 将准备好的汇编解释程序或编译系统装入计算机；
3. **compile & link**:
 1. 汇编程序或编译系统读入人工装在输入机上的穿孔卡片或穿孔带上的源程序；
 2. 执行汇编过程或编译过程，产生目标程序，并输出到目标卡片或纸带；
 3. 通过引导程序把装在输入机上的目标程序读入计算机；
4. **execute**: 启动目标程序执行，从输入机上读入人工装好的数据卡片或数据带上的数据；
5. **output**: 产生计算结果，执行结果从打印机上或卡片机上输出。



1.2.1 人工操作阶段

- 用户独占全机资源，资源利用率不高，系统效率低下
- 代码维护困难，手工操作多，浪费处理机时间
 - 1955 年的美国空军“贤者”防空系统，用了 62500 张卡片，数据量为 5M
 - 需要在一堆卡片的侧面对角线位置划一条线，来保证卡片间的顺序一致性
- 数据的输入，程序的执行、结果的输出均**联机**进行，从上机到下机的时间拉得非常长
 - 计算机不仅到执行程序，也需要读取纸带，输出结果





1.2.1 人工操作阶段

- 随着计算机硬件速度的不断提高
 - 程序的实际运行时间显著减少，人工操作时间却变化不大，导致人机矛盾不断突出
 - CPU与慢速I/O设备之间的矛盾也日益突出
- 这些现象表明计算机的使用方式急需改变



1.2.2 执行系统阶段

- 引入作业执行系统来自动完成作业流程

装入 load - 编译 compile - 连接 link - 执行 execute - 输出 output

- 联机 I/O 技术

1. 在操作员的操作下，由管理程序将这批作业从纸带或卡片机输入到磁带上
2. 管理程序自动把磁带上的第一个作业装入内存，并执行作业
3. 当上一个作业执行后，管理程序再调入磁带上的下一个作业到内存执行，直到所有作业执行完毕

- 所有工作均是由计算机独立完成，在进行数据 I/O 时就不能执行程序，反之亦然



1.2.2 执行系统阶段

• 脱机I/O技术

- 在计算机系统中同时设置一台主机和一台辅机
- 辅机仅与I/O设备打交道，不与主机连接
- 输入设备上的作业通过辅机输到磁带上，称为脱机输入
- 主机负责从磁带上把作业读入内存执行，作业完成后，主机负责把结果输出到磁带上，称为脱机输出
- 然后，由辅机把磁带上的结果信息在打印机上打印输出
- I/O工作脱离了主机，辅机和主机可以并行工作



load

compile

link

execute

output

人工
操作



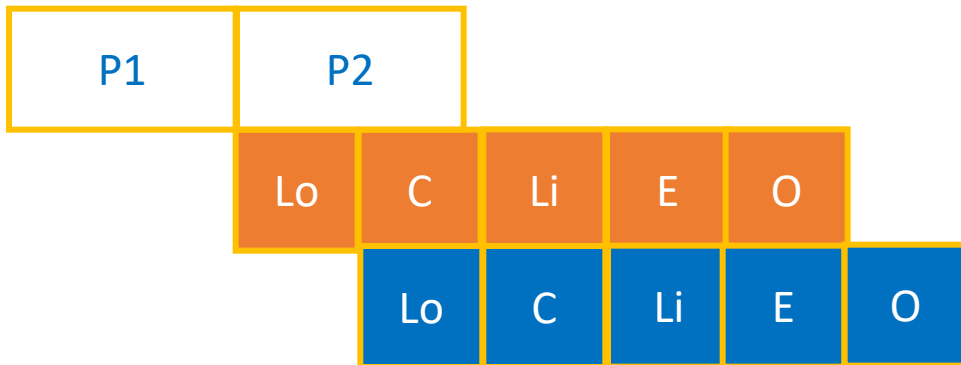
联机
I/O



脱机
I/O



多道
程序
设计





1.2.3 多道程序设计

- 允许多个程序同时进入主存储器并交替计算的方法。即计算机内存中同时存放了多道程序，它们都处于开始和结束点之间
- 从宏观上看，多道程序并发运行，它们都处于运行过程中，但都未运行结束
- 从微观上看，多道程序的执行是串行的，各道程序轮流占用CPU，交替地执行

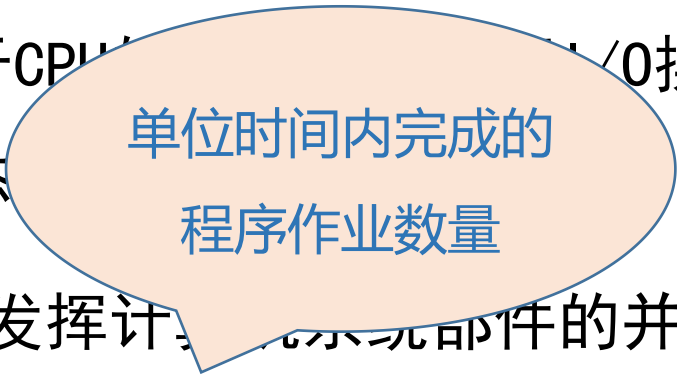


1.2.3 多道程序设计

- 多道程序技术离不开中断和通道技术的发展。
 - 中断：避免CPU因不断轮询设备而浪费时间
 - 通道：专门负责I/O的独立处理器，减轻CPU的处理压力
- 一般来说，I/O操作速度远小于CPU的速度，在进行I/O操作时，CPU处于完全的闲置状态
- 需要引入多道程序设计，充分发挥计算机系统部件的并行工作能力，提高CPU利用率和系统吞吐量



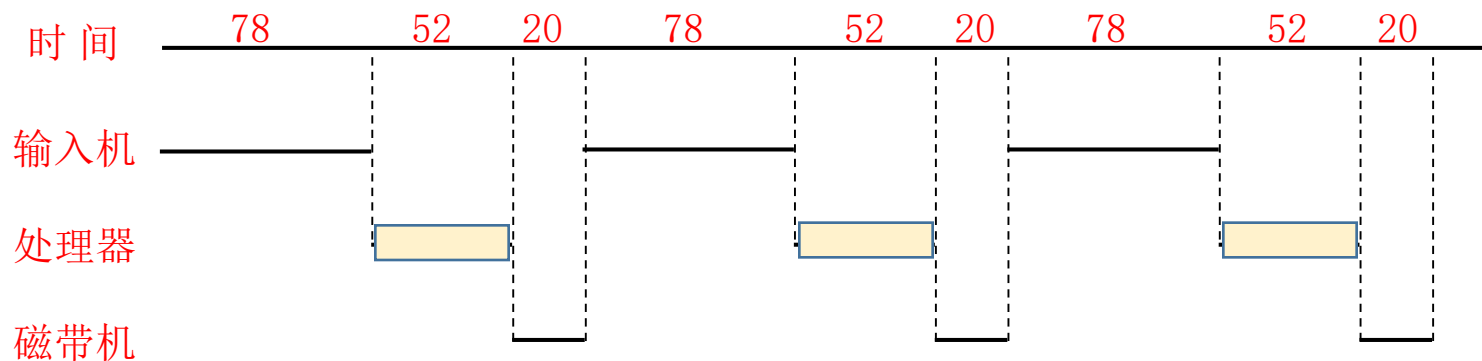
1.2.3 多道程序设计

- 多道程序技术离不开中断和通道技术的发展。
 - 中断：避免CPU因不断轮询设备而浪费时间
 - 通道：专门负责I/O的独立处理器，减轻CPU的处理压力
- 一般来说，I/O操作速度远小于CPU操作速度。当CPU等待I/O操作时，CPU处于完全的闲置状态。单位时间内完成的
程序作业数量
- 需要引入多道程序设计，充分发挥计算机各部件的并行工作能力，提高CPU利用率和系统吞吐量



1.2.3 多道程序设计

- **例：**计算数据处理问题P1，从输入机上输入500个字符（花78ms），经CPU处理52ms后，将结果2000个字符存到磁带上（花20ms），重复进行，直至输入数据全部处理完毕

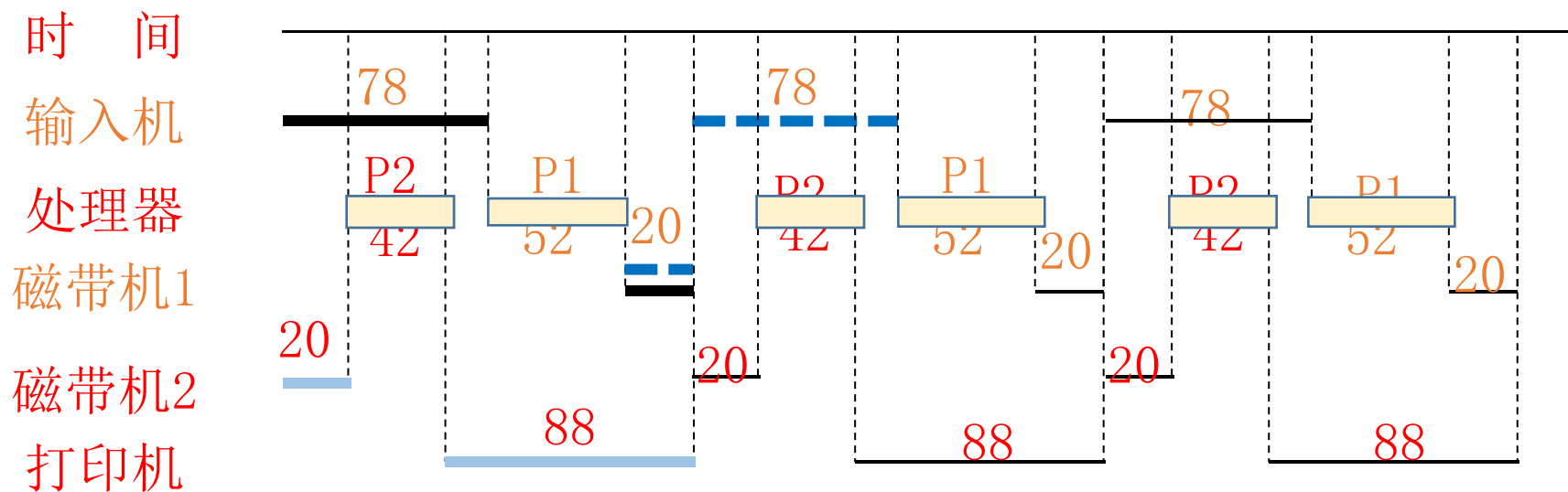


处理机的利用率为： $52 / (78 + 52 + 20) \approx 35\%$



1.2.3 多道程序设计

- 为提高效率，在计算P1的同时，计算机还接受了另一算题P2：从另一台磁带机2上输入2000个字符（花 20ms），经42 ms的处理后，从行式打印机上输出两行（约花 88ms）



处理机利用率为： $(52+42) / (78+52+20) \approx 63\%$



1.2.3 多道程序设计

- 多道程序设计利弊

- 提高了CPU的利用率
- 提高了内存和I/O设备的利用率
- 提高了系统的吞吐率
- 充分发挥了系统的并行性

- 每单道程序延长了计算时间
- 延长了作业周转时间
- 牺牲了用户的响应时间



1.2.3 多道程序设计

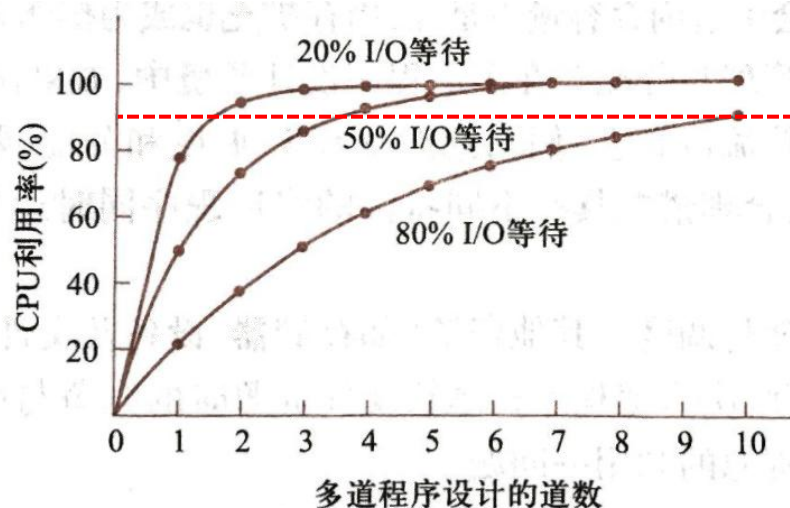
- 在进行多道程序设计时，需要关注并发道数问题
 - 并不是道数越多，效率就越高
 - 内存容量和用户响应时间等因素也影响多道程序道数的多寡
- 假设单道程序等待I/O操作时间占其运行时间的比例为 p ，当内存中有 n 道程序时，所有程序都等待I/O的概率是 p^n ，则：

$$\text{CPU利用率} = 1 - p^n$$



1.2.3 多道程序设计

- 若进程平均耗费 80% 时间等待 I/O，则均要在内存中保持 10 道进程。
- 在 1MB 内存内，若操作系统进程占据 200KB 空间，每个应用程序占据 200KB 内存，最多可容纳 4 道进程，在此情况下：



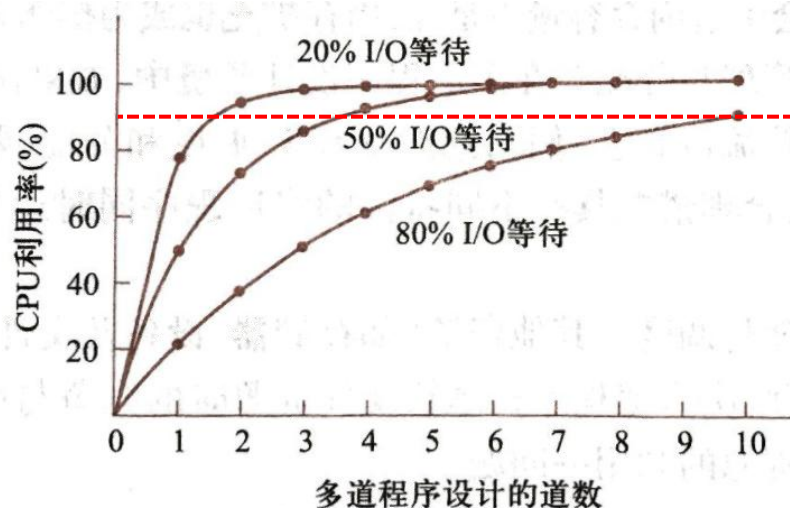
$$CPU \text{ 利用率} = 1 - 0.8^4 \approx 59\%$$



1.2.3 多道程序设计

- 若增加1MB内存，除去操作系统进程外，可容纳应用进程数量增加至9道，在此情况下CPU利用率提高了47%

I/O 等待时间、内存容量、用户等待时间等多种因素，都会对多道程序多寡产生影响。



$$CPU \text{ 利用率} = 1 - 0.8^9 \approx 87\%$$



1.2.3 多道程序设计

- 多道程序设计系统 (Multi-programming system)
- 多重处理系统 (Multi-processing system)
 - 配置多个物理处理器
 - 能够并行执行多道程序
- 多重处理系统，必须采用多道程序设计技术；而多道程序设计不一定要有多重处理系统支持。



1.2.3 多道程序设计

- 实现多道程序设计必须妥善地解决三个问题

1. 存储保护、程序浮动和内存扩充

- 多道进程间，只访问自己的区域；每个进程并不能占据内存中的固定位置，而是由操作系统进行分配，需要进行地址重定位；多个进程占据的空间会超出物理内存大小，需要辅存支持。

2. 处理器的管理和调度

- 多个进程需要轮流（或依据其它策略）来占据处理器时间。

3. 系统资源的管理和调度

- 其它系统资源也需要按一定策略来分配给不同进程使用。



1.2.3 操作系统的形成

- 中断和通道技术的发展，奠定了多道程序设计的基础
- 多道程序设计同时也需要性能更加强大的外存设备
- 20世纪60年代中期，磁盘的发明，成为了操作系统的最后一块拼图
- 操作系统显著提高了计算机资源管理水平和自动化操作效率，提供了存储管理、文件管理、分时和实时操作等功能，多道程序设计发展日臻完善。



1.2.4 操作系统发展与分类

- 操作系统的发展动力主要有以下五个方面：

1. 器件快速更新换代

- 计算机体系结构不断发展
- 内存管理支撑硬件由分页、分段机制代替了界寄存器
- 图形终端代替字符终端
- 中断、通道设施的引入
- 单处理机改进为多处理机系统
- 计算机网络的出现和发展
- 信息家电的发展等都是计算机体系结构的发展

2. 提高计算机系统资源利用率的需要

3. 让用户使用计算机越来越方便的需要

- 批处理系统发展为分时系统

4. 字符用户界面发展为图形用户界面

5. 满足用户新要求，提供给用户新服务

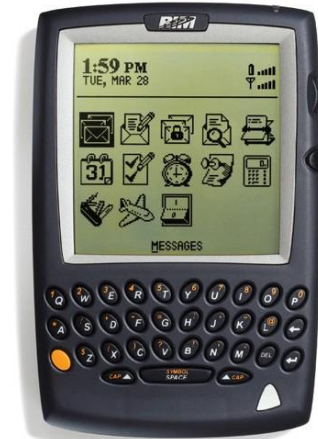
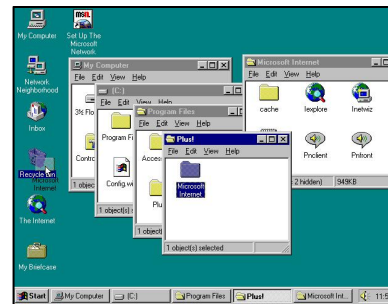
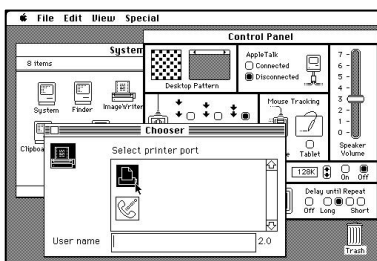
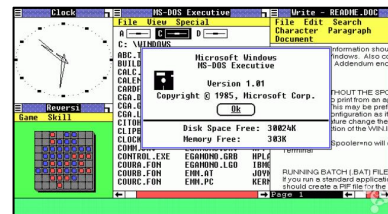
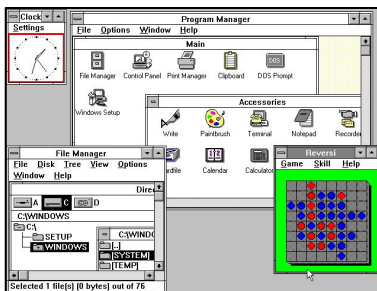
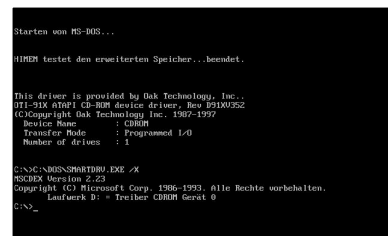
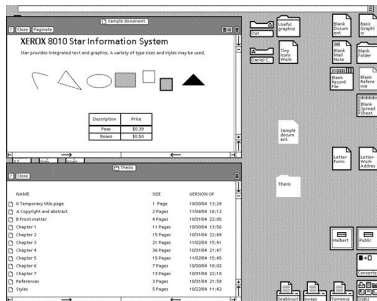


1.2.4 操作系统发展与分类

1. 批处理系统
2. 分时操作系统
3. 实时操作系统

• 进一步发展

- 微机操作系统
- 并行操作系统
- 网络操作系统
- 分布式操作系统
- 嵌入式操作系统





1.2.4 操作系统发展与分类

- 批处理操作系统 (Batch Operating System)

- 用户把要计算的应用问题编成程序，连同数据和作业说明书一起交给操作员，操作员集中一批作业，输入到计算机中。然后，由操作系统来调度和控制作业的执行。这种批量化处理作业方式的操作系统称为批处理操作系统

- 批处理系统的主要特征（优缺点）

- 用户脱机工作
- 成批处理作业
- 单/多个程序运行

Batch 如今多指
“非交互”式计算，
如.sh, .bat程序等



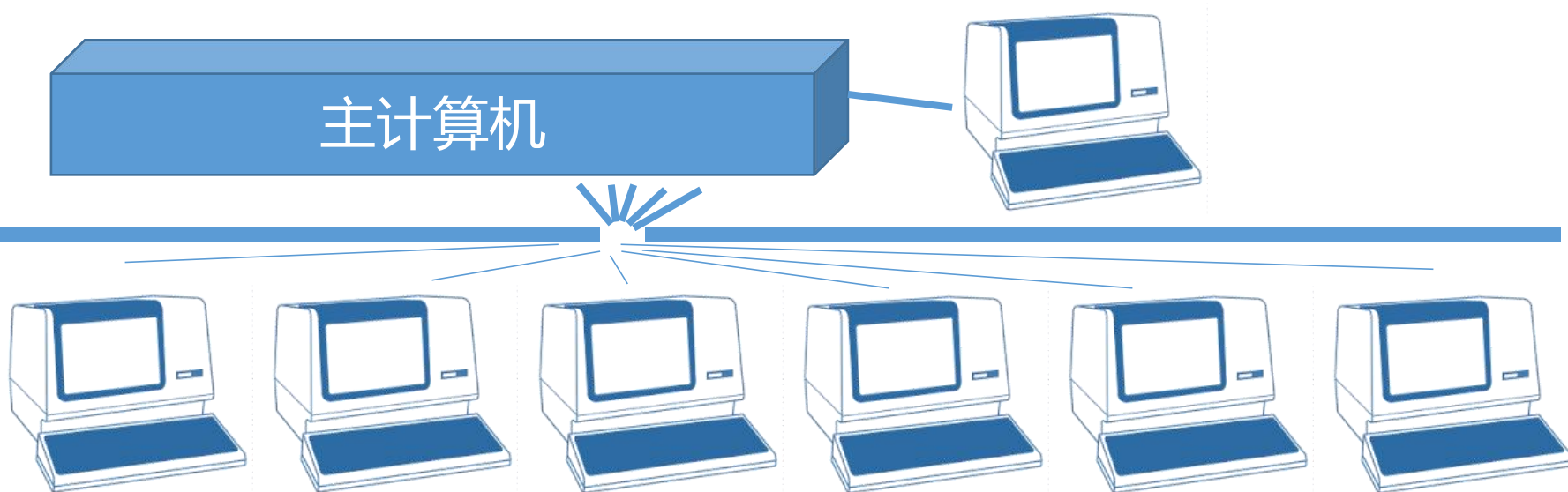
1.2.4 操作系统发展与分类

- 分时操作系统 (Time Sharing Operating System)
 - 允许多个联机用户同时使用一台计算机系统操作系统
 - 在一台主机上连接有多个终端，每个用户在各自的终端上以问答方式控制程序运行，主机中央处理器轮流为每个终端用户服务一段很短的时间，这段时间称为一个时间片，若一个终端用户的程序在一个时间片内未执行完，则挂起等待再次分到时间片时继续运行。每个用户感到自己好象独占一台计算机



1.2.4 操作系统发展与分类

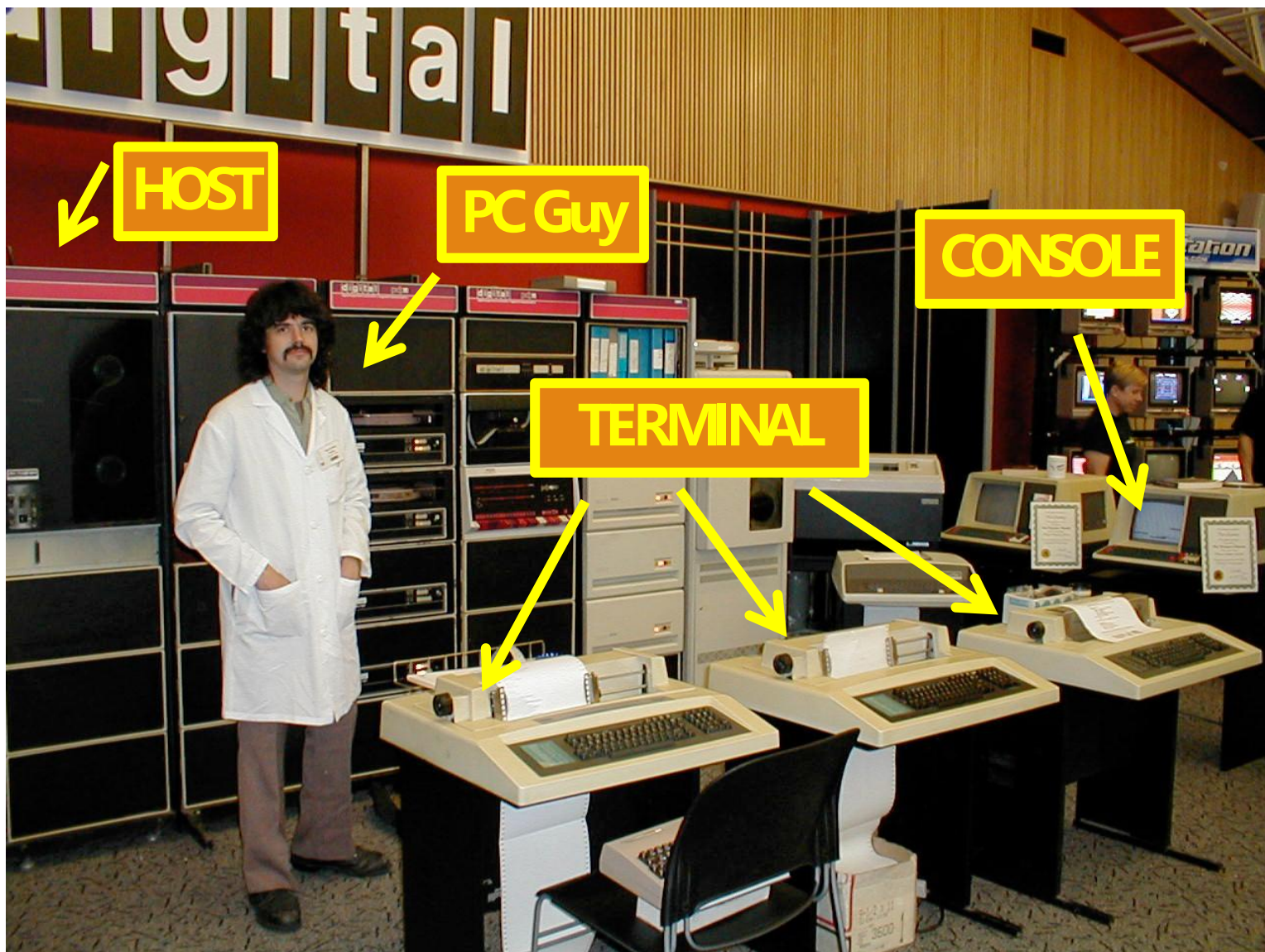
- 分时操作系统 (Time Sharing Operating System)
 - 允许多个联机用户同时使用一台计算机系统操作系统





河海大学

计算机与信息学院





1.2.4 操作系统发展与分类

- 分时操作系统的主要特性

- **同时性**：若干个终端用户同时联机使用计算机
- **独立性**：每个用户感到自己好象独占一台计算机
- **及时性**：用户发出的命令能够很快被主机响应
- **交互性**：人机交互，联机工作，方便调试、修改程序



1.2.4 操作系统发展与分类

- 分时操作系统与批处理操作系统的异同点

分时操作系统和批处理操作系统都基于多道程序设计技术

- 目标不同

- 批处理系统以提高资源利用率和作业吞吐量为目标
- 分时系统以满足多个联机用户的立即型命令的快速响应为目标

- 适应作业的性质不同

- 批处理系统适应已经调试好的大型作业
- 分时系统适应正在调试的小作业

- 资源使用率不同

- 作业控制方式不同

- 批处理由用户预先提交作业控制说明书脱机工作
- 分时系统由联机用户从键盘输入操作命令直接对作业的运行过程进行控制



1.2.4 操作系统发展与分类

- 分时系统的出现也带来了人机交互方式的改变
 - 传统分时系统是“行命令”方式 (Command Line Interface CLI)
 - 斯坦福研究所于1960年代提出：**人机交互的中心是人**
 - 70年代末，日本施乐公司提出图形化交互方式 (Graphic User Interface GUI) **WIMP**：原型机 Star
 - W: Windows，多窗口的界面
 - I: Icon，图标表示
 - M: Menu，菜单
 - P: Pointing Devices，鼠标、触控笔、手指控制 (拖拽、单击、双击)



1.2.4 操作系统发展与分类

- 分时时间片的长度选取应根据：

- 时间片长度

- 机器速度

- 用户的多少

- 响应的要求

- 时间片设得太短会导致过多的进程切换，减少实际运行用户程序的时间比，从而降低CPU的利用率

- 系统的开销

- 时间片设得太长会使小的交互型请求的响应时间变长



1.2.4 操作系统发展与分类

- **前台作业：**一般是交互型作业，用户需要频繁地与作业进行交互
- **后台作业：**一般为批处理型作业，运行时间长，且不需要用户频繁与之交互



1.2.4 操作系统发展与分类

- 实时操作系统 (Real Time Operating System)
 - 指当外界事件或数据产生时，能接收并以足够快的速度予以处理，处理的结果又能在规定时间内来控制监控的生产过程或对处理系统作出快速响应，并控制所有实时任务协调一致运行。





1.2.4 操作系统发展与分类

- 三种典型的实时系统

- **过程控制系统**：如生产过程控制系统、导弹制导系统、飞机自动驾驶系统、火炮自动控制系统
- **信息查询系统**：计算机同时从成百上千的终端接受服务请求和提问，并在短时间内作出回答和响应。如情报检索系统
- **事务处理系统**：计算机不仅要为终端用户及时作出响应，还要频繁更新系统中的文件或数据库。如银行业务系统

分时操作系统为用户提供一个通用的交互型开发运行环境

实时操作系统通常为特殊用途提供专用系统