第2章 操作系统概述

2.1 操作系统的目标和功能

2.2* 操作系统的发展史

2.3 主要成就

2.4 现代操作系统的特征

2.5 容错性

2.6 多CPU和多核OS设计因素

2.7～10 历史上的操作系统

2.1 操作系统的目标和功能

- **操作系统 Operating System：**
  - 控制用户程序执行，充当用户程序和计算机硬件之间的接口。
- 操作系统主要功能：进程管理，内存管理，设备管理，文件管理。
- 操作系统的目标：
  - 方便：使计算机易于使用，界面友好。
  - 有效：提高计算机系统的资源利用率。
  - 扩展能力：可扩展、兼容、移植新的功能。

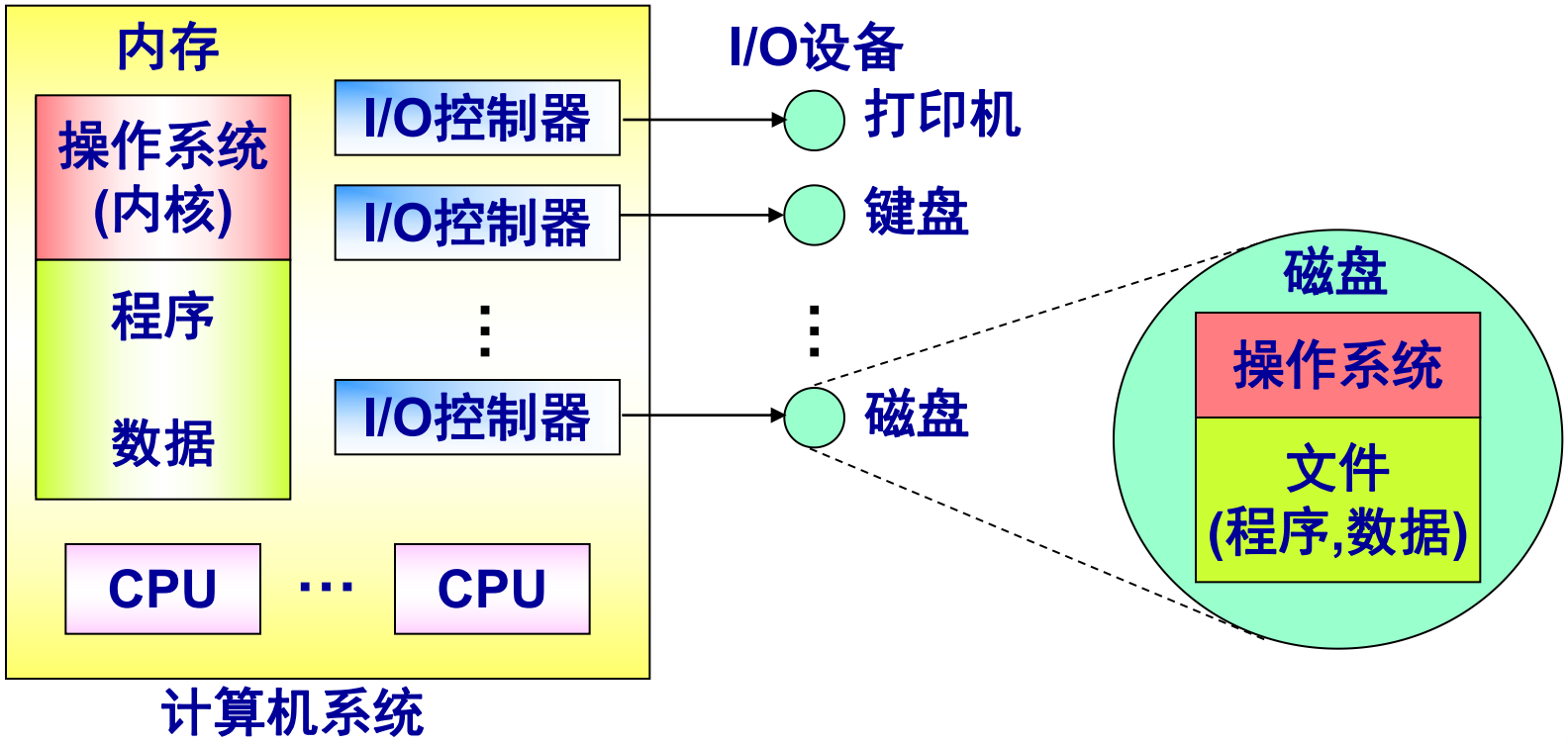
2.1.1 作为用户/计算机接口的操作系统－方便

- OS为用户提供各种服务，方便用户使用计算机：
  - 程序运行：分配内存/文件/设备，调度CPU等；
  - I/O设备访问：隐藏设备细节，提供统一的API接口；
  - 文件访问控制：按名透明存取、共享/保护文件；
  - 系统访问：共享/保护资源，资源竞争时的冲突；
  - 错误检测和响应：响应软/硬件错误，终止/重试/报告。
  - 记账：收集和监控性能参数、资源利用率等。

向用户屏蔽软件/硬件细节，提供方便的接口

2.1.2 作为资源管理器的操作系统－有效

- OS提高软硬件资源的利用率。OS控制和决定：
  - 各程序的执行时机、能在CPU上运行多长时间、分配多大内存、何时使用I/O设备、对文件的访问等等。



2.1.3 操作系统的易扩展性—扩展能力

- OS需要不断改进和发展：
  - 新型硬件及硬件升级、新的服务要求、纠正错误打补丁等，都需要更新和扩展操作系统的设计。
- 为了便于扩展，构造操作系统时应采用模块化结构，清晰定义模块间接口，并备有说明文档。

Question:

1、操作系统在计算机系统中位于_____之间。

A. CPU和用户　　B. CPU和设备
C. 硬件和用户　　D. 硬件和软件

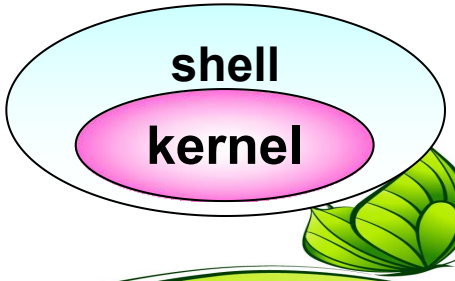
2、计算机开机后，OS被加载到_____。

A. BIOS　B. ROM　C. EPROM　D. RAM

3、OS不管理下列哪种资源？_____。

A. CPU　B. 内存　C. 外存　D. cache

4、操作环境不是操作系统，对吗？

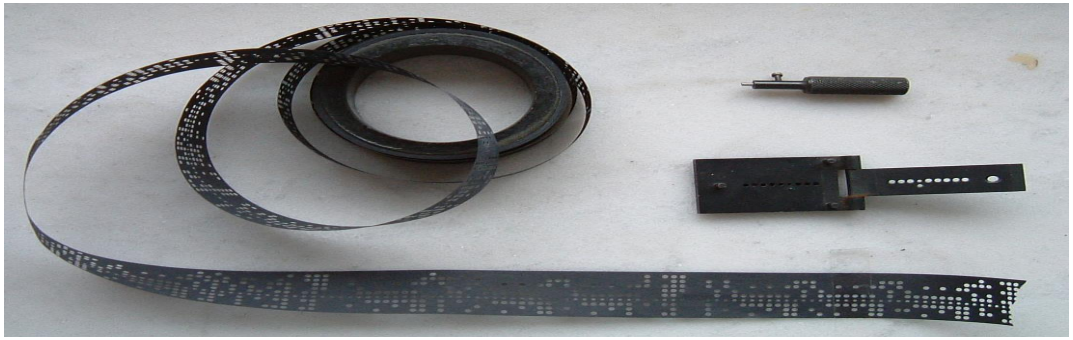


2.2 操作系统的发展

2.2.1 串行处理－没有操作系统

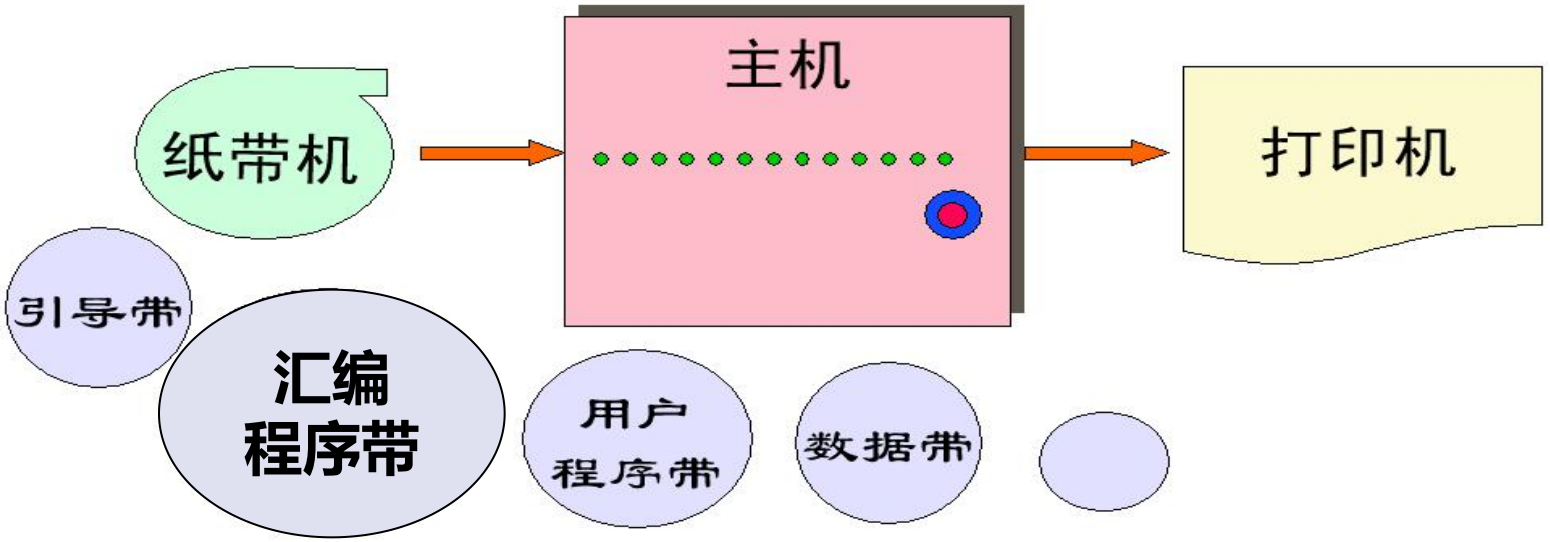
- 1946～50年代中期的人工操作阶段——无OS
  - 编程语言：机器语言、汇编语言；
  - 输入设备：纸带或卡片机；
  - 输出设备：显示灯，打印机；
  - 用户既是程序员，又是操作员；
  - 用户在预约机时内独占全机，资源利用率低；

纸带



2.2.1 串行处理—没有操作系统

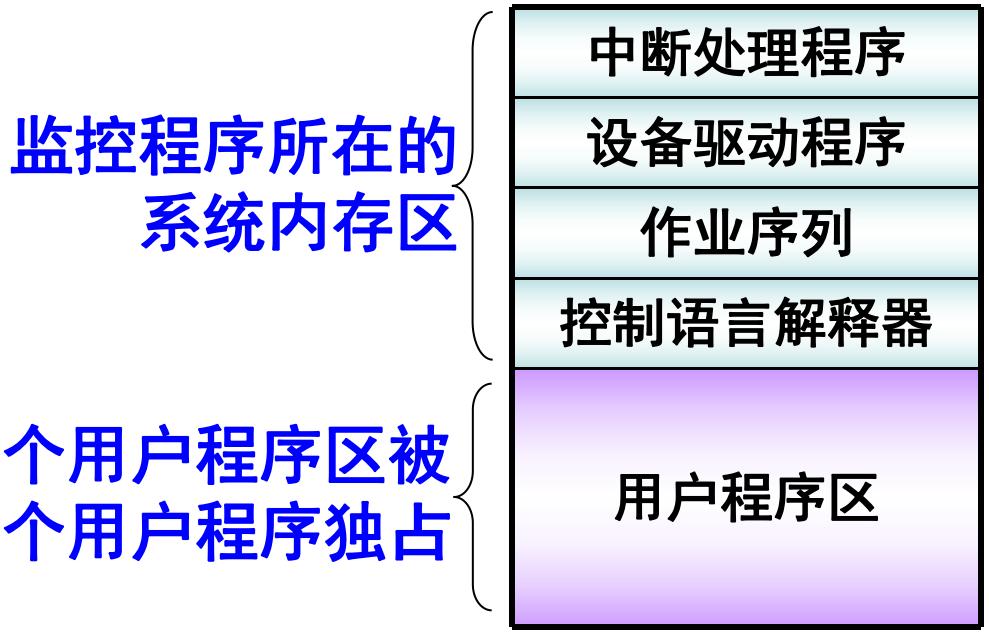
- 人工操作的缺陷：
  - CPU等待用户：计算前，手工装入纸带或卡片；计算完成后，手工卸取纸带或卡片；CPU利用率低。



突出矛盾：人工操作速度和计算机速度的差异

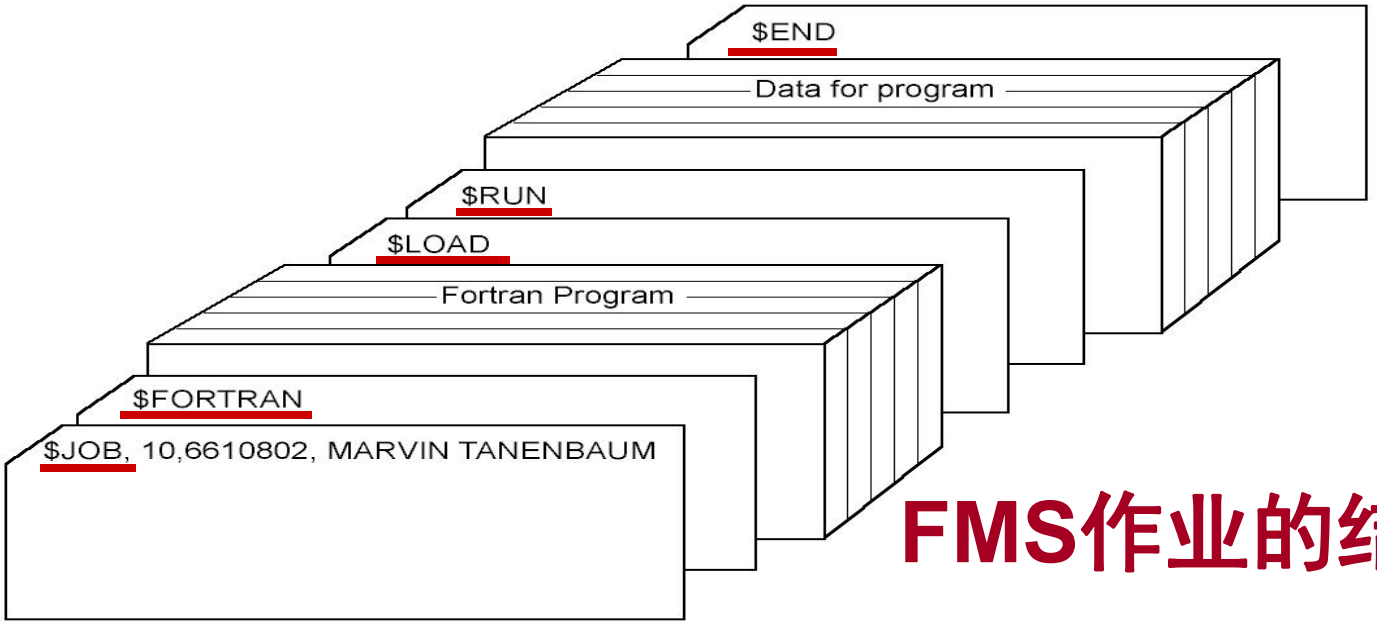
2.2.2 简单批处理系统－监控程序Monitor

- 50年代中～60年代中，监控程序（雏形OS）
- 为了减少手工操作而导致的CPU空闲，实现程序的自动衔接运行。监控程序使磁带上的一批作业自动、顺序地逐个运行。
- 内存中只有一个作业，故称“单道批处理”。



2.2.2 简单批处理系统－监控程序Monitor

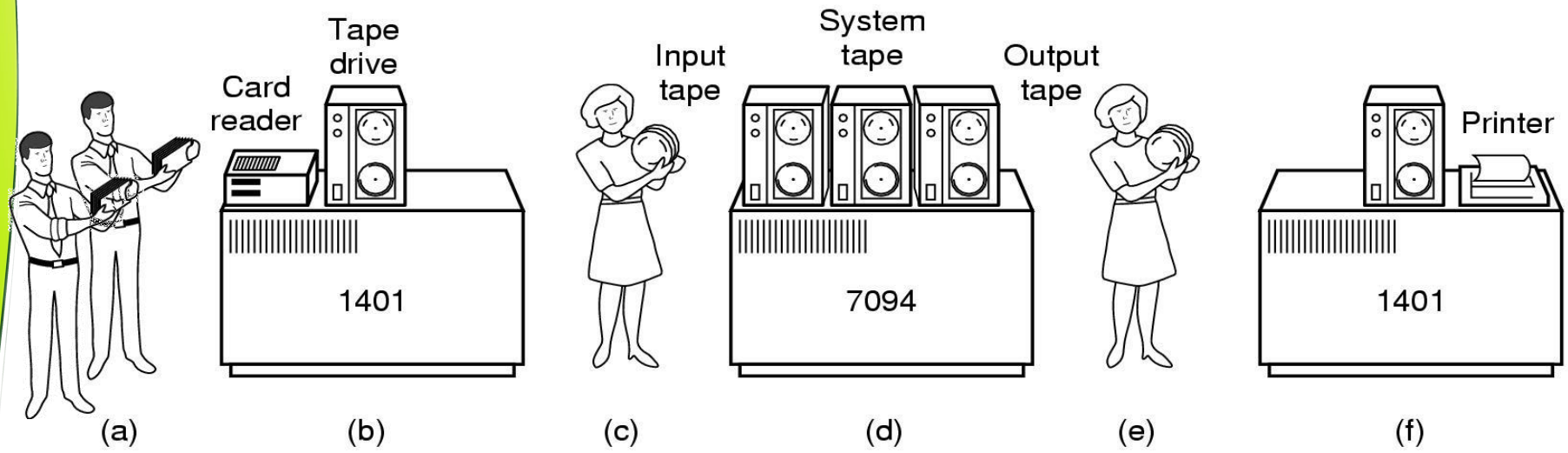
- 批处理系统中，作业的组成：
  - 用户程序，数据，作业说明书
  - 作业说明书：（作业控制语言JCL）
    $ ABC, JOB(918, 001), Chen-Ling, CLASS=A,
    TIME=(5, 20), REGION=15K, PRIORITY=5



FMS作业的结构

2.2.2 简单批处理系统－脱机I/O

- 批处理系统的工作方式——脱机输入输出



程序员把程序卡片拿到卫星机上，读入到“输入磁带”。

操作员把“输入磁带”放进主机，作业在Monitor控制下自动逐个(单道)运行，运算结果写到“输出磁带”上。

“输出磁带”送到卫星机上去打印。

2.2.3 多道批处理系统

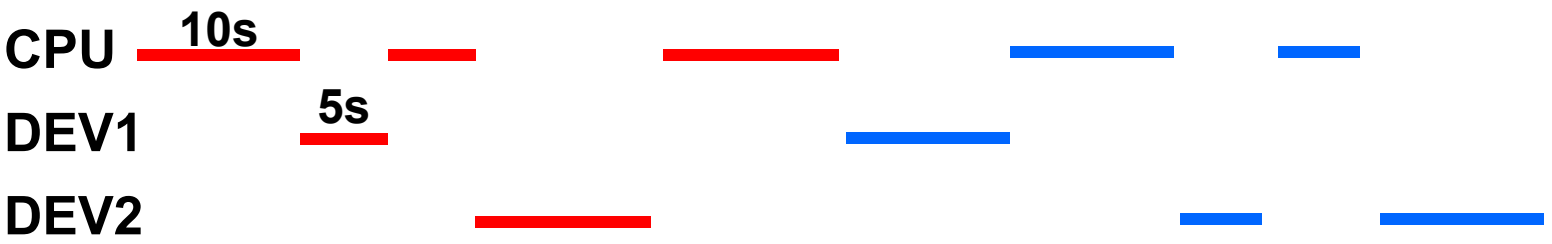
- **多道程序设计**：60年代中～70年代中，由于中断、磁盘、DMA等的引入，使得内存中可以同时存在多个作业，并发运行。
  - 并发concurrency：多个事件在同一时间间隔内发生。
    并行parallelism：多个事件在同一时刻发生。
- 当正执行的作业需要进行I/O时，可以让出CPU给内存中的其它作业去运行，以**提高资源利用率**。
- **多道批处理OS**：
  - 减少了因CPU与I/O设备串行工作导致的CPU空闲。CPU与I/O设备并行工作，资源利用率高，吞吐率高。
  - 周转时间长，用户交互性差：整个作业完成后或中间出错时，才与用户交互，不利于调试和修改。

2.2.3 多道批处理系统

- 多道程序设计提高资源利用率和吞吐率的例1：

设程序A用 —— 表示，程序B用 —— 表示。

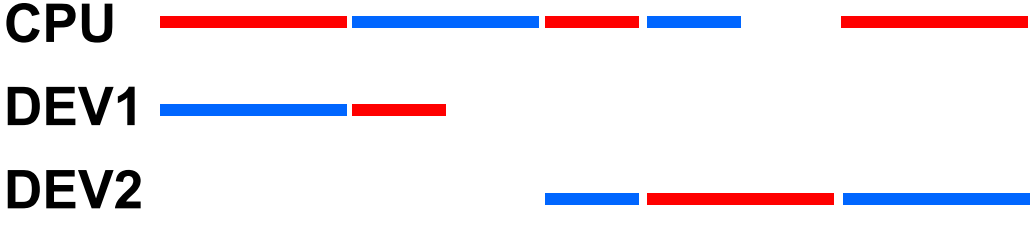
单道顺序执行时：



吞吐量：平均40s完成1个作业。资源利用率:

CPU=40/80=50% DEV1=15/80=18.75% DEV2=25/80=31.25%

多道并发执行时：设A、B优先级相同，不能互相抢占CPU



CPU=40/45=89%

DEV1=15/45=33%

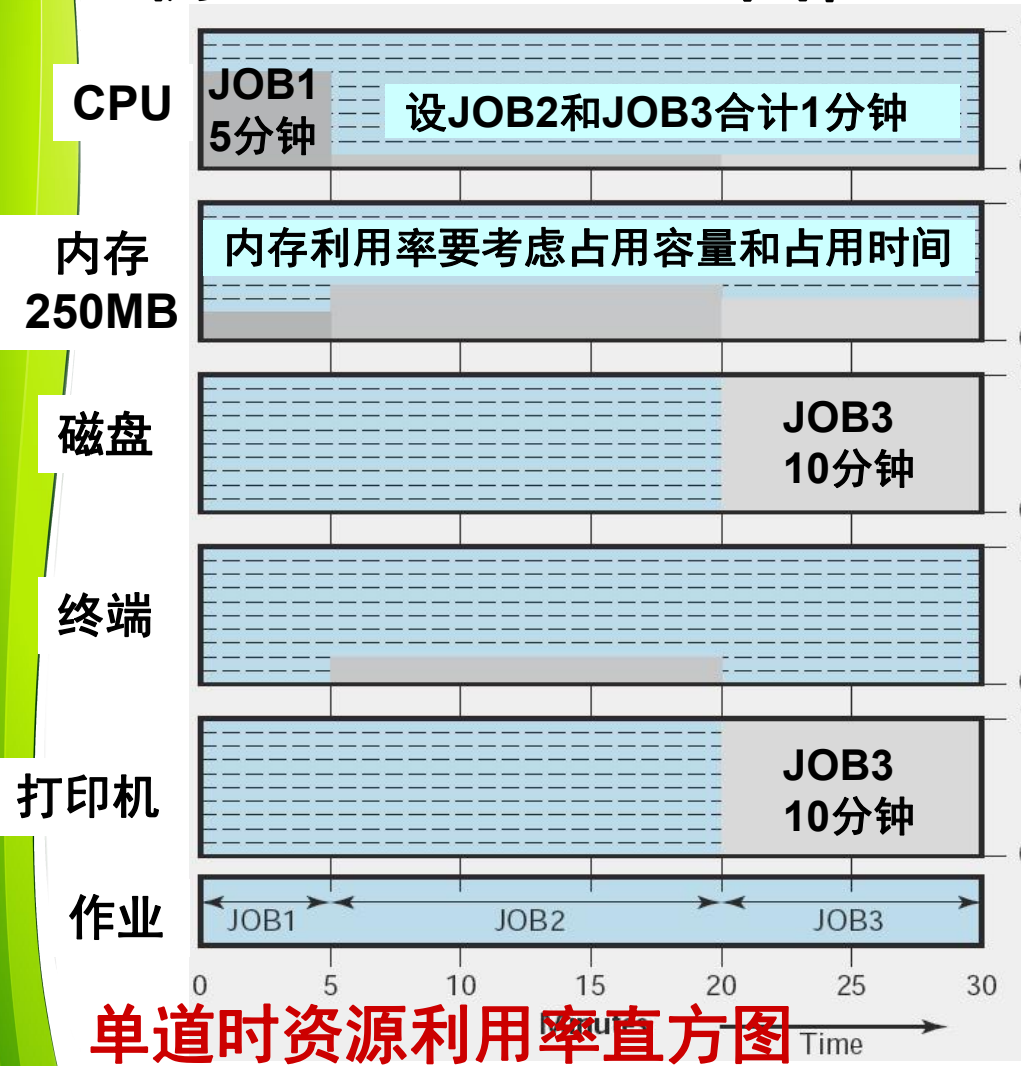
DEV2=25/45=56%

吞吐量：45s完成2个作业。

2.2.3 多道批处理系统

	JOB1	JOB2	JOB3
作业类型	计算	I/O	I/O
持续时间	5分钟	15分钟	10分钟
所需内存	50MB	100MB	75MB
需要磁盘	否	否	是
需要终端	否	是	否
打印机	否	否	是

- 例2：P37~38。3个作业：



单道时资源利用率直方图

利用率及其它	单道程序
CPU	20%
内存	33%
磁盘	33%
打印机	33%
总共运行时间	30分钟
吞吐率	6作业/小时
平均周转时间	18分钟

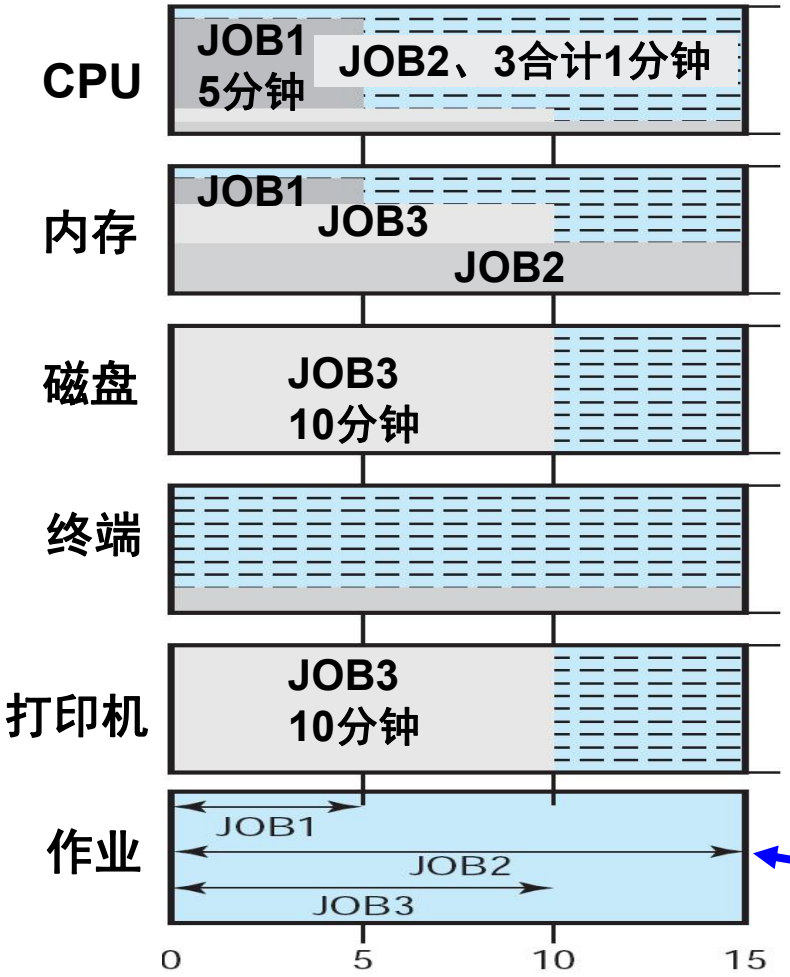
内存总容量250MB。单道时内存利用率为：

$$\frac{50}{250}\times\frac{5}{30}+\frac{100}{250}\times\frac{15}{30}+\frac{75}{250}\times\frac{10}{30}=33\%$$

单道时，各作业的周转(响应)时间为：
JOB1－5分钟，JOB2－20分钟，
JOB3－30分钟，故平均18分钟。

2.2.3 多道批处理系统

- 例2：P41~42。3个作业：此多道例课下自学



多道时资源利用率直方图

利用率及其它	单道程序	多道程序
CPU	20%	40%
内存	33%	67%
磁盘	33%	67%
打印机	33%	67%
总共运行时间	30分钟	15分钟
吞吐率	6作业/小时	12作业/小时
平均周转时间	18分钟	10分钟

多道时，各作业交替穿插执行，全部执行完毕需15分钟。

内存总容量250MB。多道时内存利用率为：

$$\frac{50}{250}\times\frac{5}{15}+\frac{100}{250}\times\frac{15}{15}+\frac{75}{250}\times\frac{10}{15}=67\%$$

多道时，各作业的周转(响应)时间为：JOB1－5分钟，JOB2－15分钟，JOB3－10分钟，故平均10分钟。

Question:

- 设CPU和I/O设备能并行工作，作业优先级从高到低依次为 A、B、C。优先级高的作业可以抢占优先级低的作业的CPU，但不可抢占I/O设备。运行轨迹：

A：CPU 20ms，I/O 30ms，CPU 10ms

B：CPU 40ms，I/O 20ms，CPU 10ms

C：CPU 10ms，I/O 30ms，CPU 20ms，I/O 20ms

- 求多道并发运行时的CPU利用率。

多道时按ABC优先级递减并发运行：

CPU: A 20 B 30 A10 B10 C10 B 10 C 20

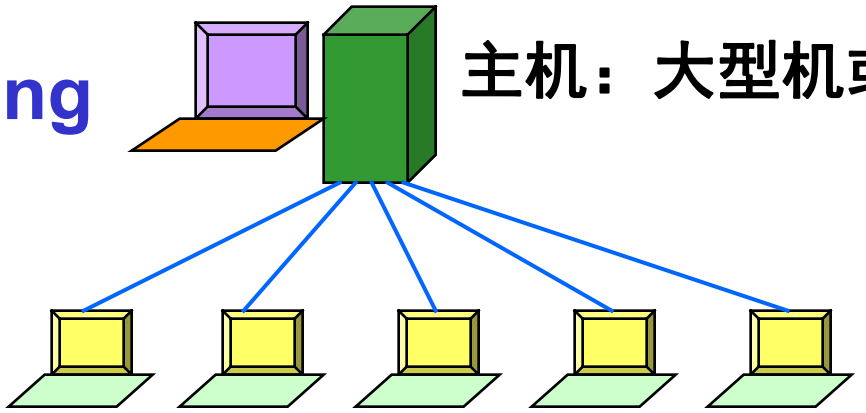
I/O: A 30 B 20 C 30 C 20

总运行160ms，CPU利用率＝110/160＝68.8%

2.2.4 分时系统－70年代中期至今

- **分时OS**：把CPU时间分割成**时间片**，每个用户依次**轮流使用**时间片。
  - 多个用户通过终端与主机交互。他们的作业按照**时间片轮转**运行。作业的**响应时间**短。

Time-Sharing System



分时：为了满足用户交互和及时响应的要求。

2.2.4 分时系统

- 批处理OS和分时OS的比较：

	批处理OS	分时OS
主要目标	提高资源利用率	减小响应时间
OS指令源	作业控制语言， 作业中的命令	终端输入的命令

- 另外：对于实时控制系统和实时信息处理系统，需要专用的**实时操作系统**（Real-Time OS）：
  - 严格的时间限制和及时响应；
  - 高可靠性：如采用冗余的硬件/软件。
- 兼有几种OS特征的**通用OS**中的前台和后台处理：
  - 前台（分时/实时）& 后台（批处理）

Question:

OS的主要性能指标有_____吞吐率_____和_____资源利用率_____。

如果一个OS具有很强的交互性，可同时供多个用户使用，但时间响应不太及时，则属于_____分时_____操作系统；

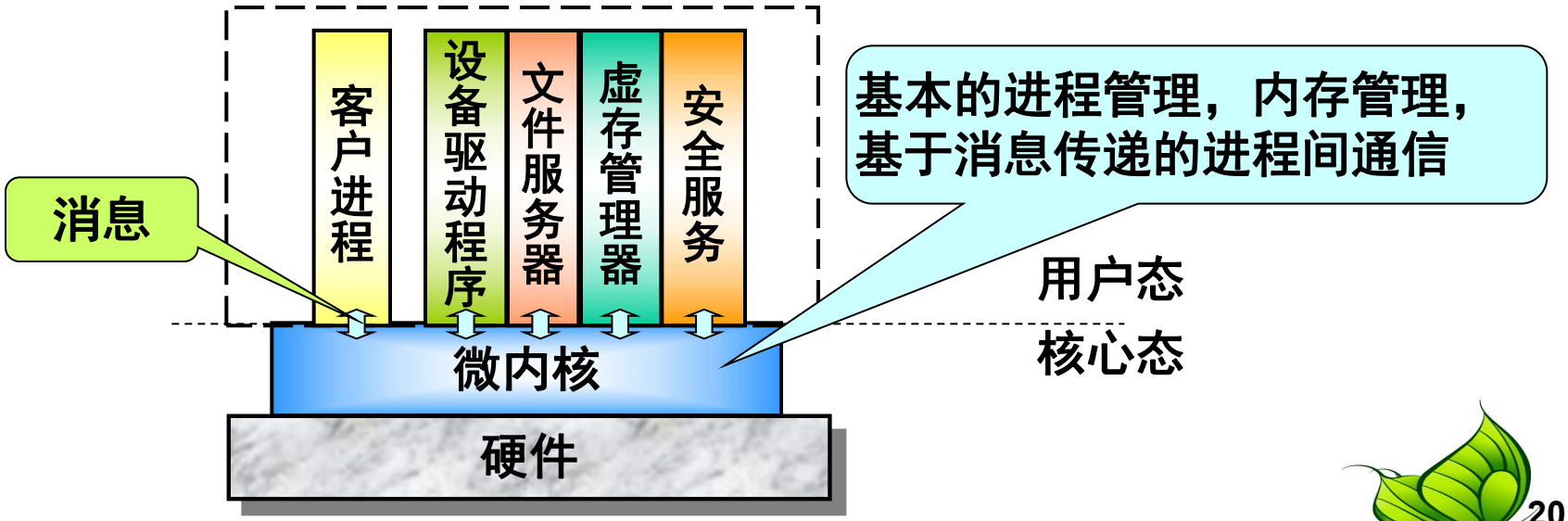
如果OS可靠，时间响应很及时但仅有简单的交互能力，则属于_____实时_____操作系统；

如果OS在用户提交作业后，不提供交互能力，它所追求的是计算机资源的高利用率，大吞吐率和作业流程的自动化，则属于_____批处理_____操作系统。

2.3 主要成就（略）

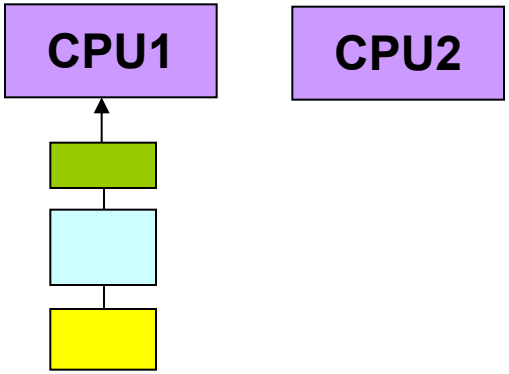
2.4 现代操作系统的特征－微内核

- 目前，大多数OS是单体内核结构：如 Linux
  - 内核kernel是OS的核心，执行进程/内存/设备/文件管理功能。在核心态下执行特权指令，能访问全部设备和内存空间。
- 微内核 microkernel：如 Mach, Apple MacOS X
  - 现代OS的趋势，将大部分传统OS内核代码分离出来放在用户层次上，只留一个最小核心运行在核心态。

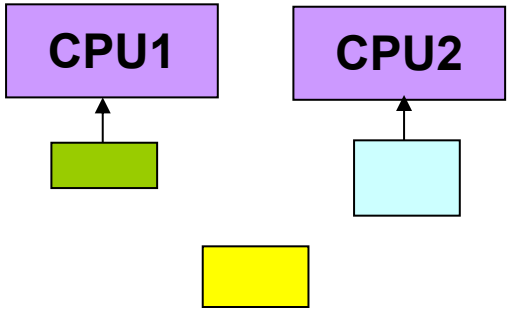


2.4 现代操作系统的特征－多线程

- **多线程 multithreading**：将一个进程划分为可以同时运行的多个线程，每个线程并发执行程序的一个模块。可利用多处理器结构。
- 多线程进程适合于执行多个独立的、不需要串行处理的任务。与进程切换相比，线程间切换开销更小。
  - 如：数据库服务器/网络服务器为每个远程客户请求建立一个线程，而不是一个进程，可以减小开销。



单线程进程：多个模块串行执行



多线程进程：多个模块并发/并行执行

2.4 现代操作系统的特征－对称多处理

- **对称多处理**：OS可调度进程或线程到所有CPU上运行。显然，多个CPU可提高性能和可用性。

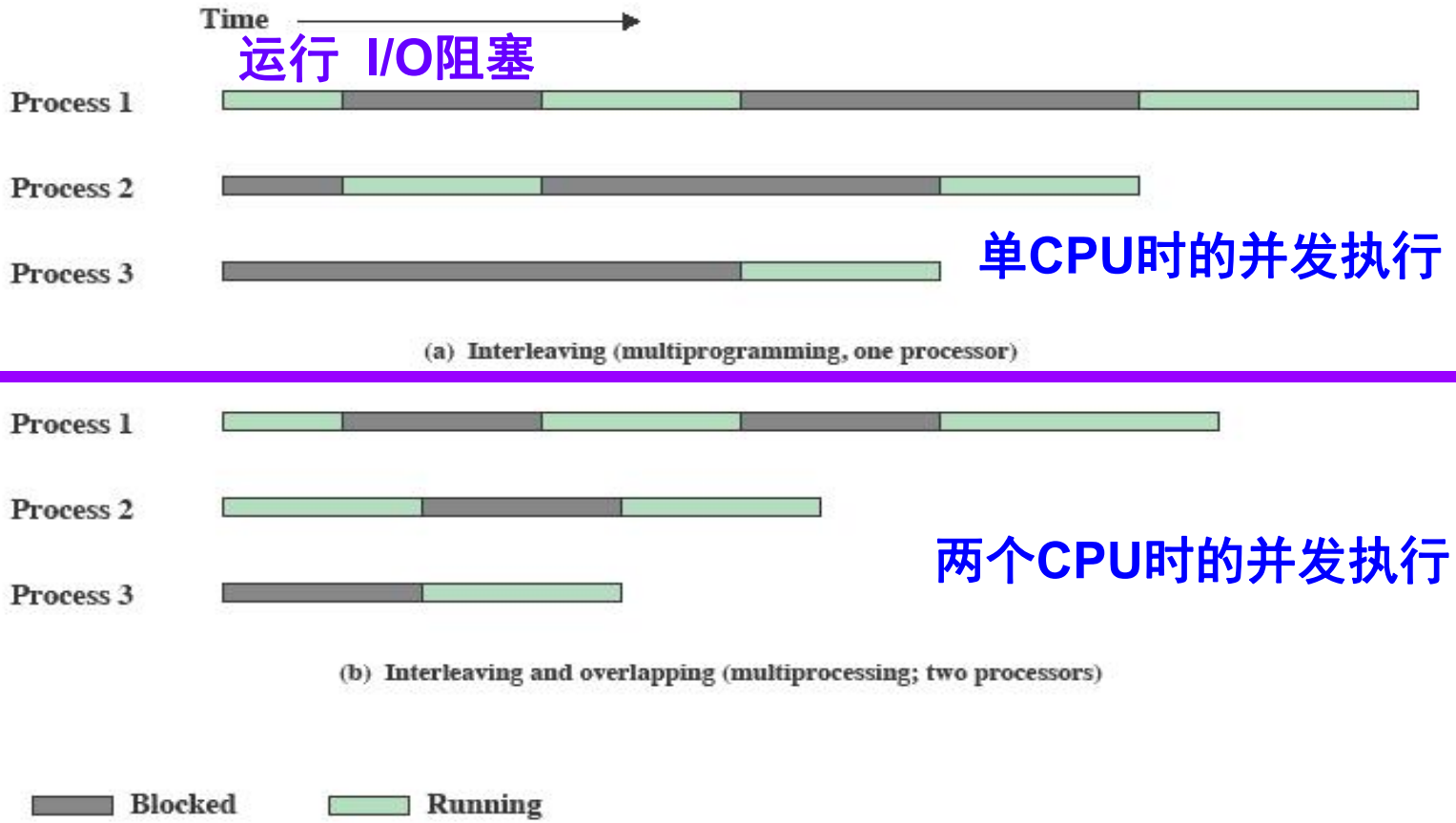
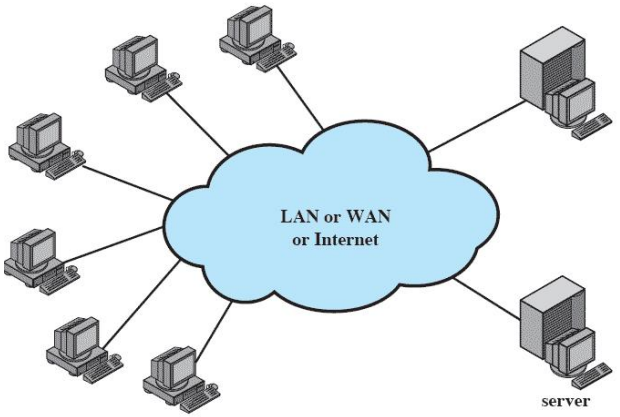


Figure 2.12 Multiprogramming and Multiprocessing

2.4 现代操作系统的特征－分布式OS

- **分布式系统(集群cluster，云)**：由多个同构或异构的高性能PC或工作站作为节点，通过高速的互连网络连接。
  - “**单计算机系统映像**”：分布式OS中的多计算机、资源分配、进程迁移、消息传递等对用户是透明的。用户看到的是一台“虚拟的单处理机”，用户不知道自己的程序在哪一节点上运行或正在访问哪一节点的文件。
  - 可靠：若干CPU故障不影响整个系统运行。
  - 资源共享：可共享远程节点文件、硬件设备等所有资源。
  - 加快运算速度：将运算分解为子运算在不同节点上并发运行，进程迁移，负载均衡。
  - 缺点：分布式软件的设计和实现困难；
    依赖于网络通信的质量；
    数据共享导致安全问题。



2.5 容错性— 2.5.1 基本概念

- **容错性**：发生软件/硬件错误时，系统能够继续正常运行的能力。改进容错性可以提高系统可靠性。
- **可靠性**：单位时间内系统正确运行的概率。
  - 平均失效（无故障）时间**MTTF**：Mean Time To Failure
    正常运行到下一次发生故障的平均时间。
  - 平均修复时间**MTTR**：Mean Time To Repair
    修复故障所花费的平均时间。
- **可用性**：系统能够有效服务用户的时间比例。宕机时系统不可用。**可用性A=MTTF/(MTTF+MTTR)**

Question:

甲系统：正常运行 修复 正常运行 修复

乙系统：

问：甲、乙系统哪个可靠性更好？ 可用性谁更好？

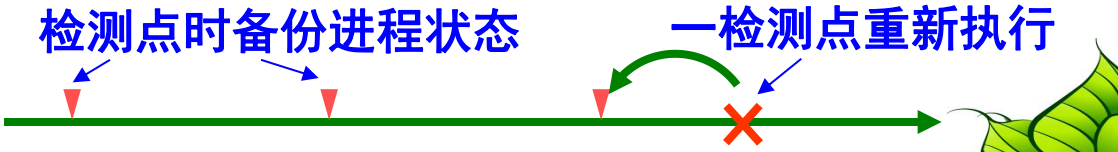
Question: 为什么说对称多处理SMP系统可用性好？

2.5.2 错误—不正确的硬件或软件状态

- 冗余可以实现容错性：
  - 空间（物理）冗余：多个硬件双工工作。
  - 时间冗余：检测到错误时重复执行操作。
  - 信息冗余：备份数据或加入校验码信息。

2.5.3 操作系统机制

- 为了提高容错性，OS采用的技术：
  - 进程隔离：多个进程访问内存、文件互不影响。
  - 并发控制：进程同步、互斥、死锁。
  - 虚拟机：用虚拟机软件（VMware、Virtual PC等）在一个硬件平台上同时运行多个OS。
  - 检测点和回滚：



2.6 针对多CPU和多核的OS设计考虑因素

- 对于对称多处理器SMP系统和多核系统：
  - 利用其潜在的并行能力：每个核或CPU并行执行指令、多进程或多线程。
  - 同时在多个CPU或核上运行多个活跃进程或线程，它们对内存、设备、内核代码和数据的同步互斥访问。

历史上的操作系统

- FMS和IBSYS（IBM为7090/7094配备的OS）
- OS/360（IBM为360系列机配备的OS）
- CTSS（第一个分时OS）
- MULTICS
- UNIX类、Linux
- MS-DOS、Windows 系列
- Macintosh Mac OS（苹果台式机OS）
- Mach（微内核OS的代表）
- OS/390、z/OS（现代IBM大型机OS）
- Android、iOS（嵌入式OS的代表）

批处理时代的操作系统

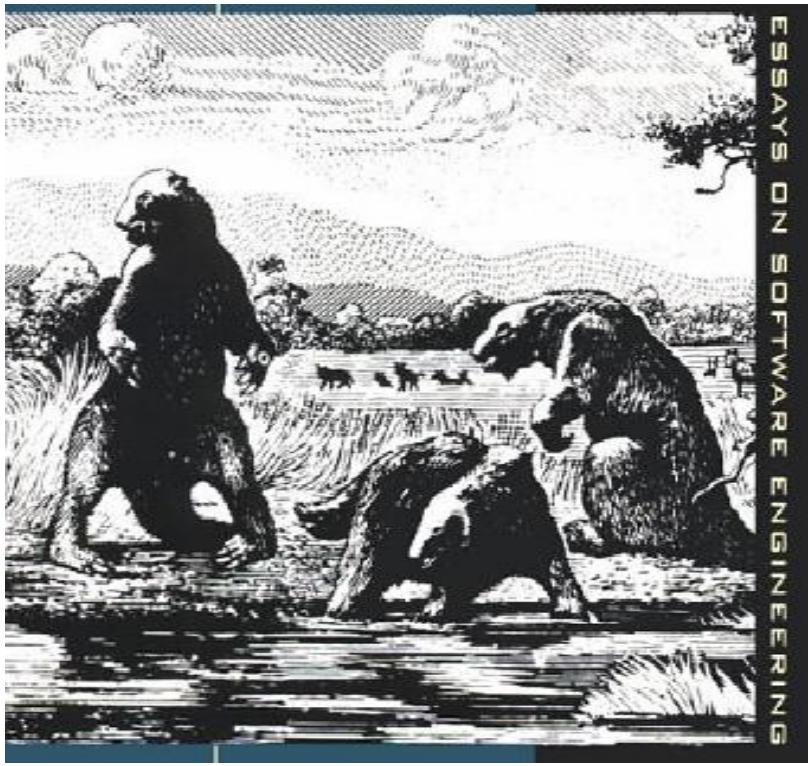
- FMS（FORTRAN Monitor System，FORTRAN监控系统）
- IBSYS（IBM为7090/7094机配备的操作系统）
- 这些操作系统由监控程序，特权指令，存储保护和简单的单道批处理构成。

1964年，IBM宣布推出System/360计算机系统(第一个采用小规模集成电路的主流机型)。多道批处理OS

由于该系列所有计算机都有相同的体系结构和指令集，在理论上，为某一型号编写的程序可以在其他型号机器上运行。（兼容&移植）

IBM OS/360—庞大的软件怪兽

- OS/360自身占据了大量存储空间和一半CPU时间；
- 数千名程序员写的数百万行汇编语言代码中，有成千上万处错误；
- IBM不断发行新版本试图更正这些错误。但每个新版本在更正老错误的同时，又引入新错误；
- 随着时间的流逝，错误的数量大致保持不变。



软件危机

Fred Brooks 软件工程

——1999年图灵奖

第一个分时操作系统CTSS

- 1961年，第一个分时系统（CTSS）由MIT的Fernando Corbato等在一改装的IBM 7090/94机上开发成功（有32个交互式用户）。
- 1962年，Manchester大学的Atlas计算机投入运行，第一个有虚拟存储器（virtual memory）和内存页面调度（paging）的机器。
- 由于CTSS和MULTICS，Fernando Corbato ——1990年图灵奖

MULTICS的灾难

- 1965年，在ARPA支持下，MIT、贝尔实验室和通用电气公司决定开发称作 MULTICS 的“公用计算服务系统”，希望能够同时支持整个波士顿所有的分时用户。
- MULTICS研制难度超出所有人的预料，1969年4月贝尔实验室退出，通用电气公司也退出。
- 最终，MIT坚持下来，MULTICS成功运行，成为商业产品（通用汽车、福特、美国国家安全局等）。运行MULTICS的计算机系统在九十年代陆续被关闭（加拿大国防部于2000年10月30日17:08）。
- MULTICS的意义：引入了许多现代操作系统的概念雏形，对随后的操作系统，特别是UNIX的成功有着巨大的影响。

小型计算机和UNIX的成功

- 1969年，在贝尔实验室退出MULTICS研制项目后，Ken Thompson 和 Dennis M. Ritchie 想申请经费买计算机从事操作系统研究，但多次申请得不到批准，项目无着落，他们在一台无人用的PDP-7上，重新摆弄原先在MULTICS项目上设计的“空间旅行”游戏。
- 为了使游戏能够在PDP-7上顺利运行，他们用汇编语言陆续开发了浮点运算软件包、显示驱动软件，设计了文件系统、实用程序、shell 等。
- 到了1970年，在一切完成后，他们给新系统起了个同MULTICS发音相近的名字——UNIX。
- 1969年，Ken设计了B语言。1972年，Dennis在B基础上设计了 C语言。——1983年图灵奖
- 1973年，UNIX用C语言重写。C和UNIX诞生。

经久不衰的UNIX

- UNIX是现代操作系统的代表。其安全性、可靠性以及强大的计算能力赢得广大用户的信赖。
- 促使UNIX系统成功的因素：
  - 首先，UNIX是用C语言编写，易开发可移植。
  - 第二，系统源代码效率高，容易适应特殊的需求。
  - 最重要的，它是通用的多用户多任务的分时OS。
- 两个版本系列：UNIX变种
  - AT&T System V：Sun Solaris，IBM AIX……
  - BSD（Berkeley Software Distribution）：FreeBSD，OpenBSD，Mac OS X……

PC机和脱颖而出的微软及MS DOS

- 1980年，IBM决定尽快生产出个人计算机——微机，但没有操作系统不行，微软公司毛遂自荐。
- 在关键时刻，开发新OS时间和人手上已经不可能，微软找到西雅图计算机产品公司，用5万美元购买了西雅图的QDOS操作系统。当时西雅图公司并不知道QDOS将被转卖给IBM，否则历史将会怎样演变，谁也无法知晓。
- IBM在1981年推出 PC机，宣布了 DOS。

拯救苹果公司的Macintosh（MAC OS）

- 1977年，苹果公司推出Apple微机。但1981年IBM PC机的巨大成功使得Apple销量落后于蓝色巨人。
- 1979年，施乐公司允许乔布斯考察Palo Alto。令苹果吃惊的是，施乐公司在拥有这些宝贵技术的同时竟然什么也没有做！
- 于是，苹果决定立即开发采用这些新技术的个人计算机。

施乐Palo Alto研究中心——70年代的计算机研究思想库

- 世界上第一台个人计算机Alto，1972年在这里出现
- 图形界面、手持鼠标、面向对象程序设计、微机网络、桌面出版、激光打印 等等先进概念和技术的原型都首次出现在这里

MAC OS和带鼠标的新型个人计算机

- 1984年，人们看到一则广告：“What was that?”和对Macintosh的介绍，这是配有图形界面操作系统MAC OS和鼠标的新型个人计算机。
- MAC机一上市立即在市场上获得极大的成功。
- 苹果公司又开始向前发展。正是Mac先进的图形界面技术，超前 IBM PC 机若干年，造就了一批苹果的忠实追随者。

- 苹果桌面一体机 iMac 和笔记本 MacBook 的操作系统：OS X （基于UNIX）
- 苹果移动设备的操作系统：iOS（基于FreeBSD和Mach的Darwin内核）

一波三折的微软Windows操作系统

- 1984年，PC机竞争厂家（Apple）的图形界面相关产品上市。
- 面对市场压力，比尔.盖茨宣布推出基于DOS的Windows操作系统，但直到1985年11月20日，Windows 1.0才正式上市，但反响不如意。
- 1990年，推出Windows 3.0，赢得市场认可；1993年，推出Windows NT，摆脱了DOS平台。
- Windows 95/98、ME、2000、XP(2001)、Vista(2007)、7(2009)、8(2011) 、10(2015) 、Azure(面向云计算)……

微内核的代表—Mach操作系统

- 1984年，在ARPA支持下，卡耐基.梅隆大学开始开发Mach，希望Mach能与UNIX兼容。
- 1986年，第一个版本。1988年的Mach 2.5版包含了大量的BSD UNIX的代码。
- 1989年，去掉了所有的BSD UNIX的代码，剩下了一个纯的Mach微内核，即Mach 3.0版本。
- 在Mach的基础上，有不少用于微处理器、多处理器以及超级计算机的操作系统和实时嵌入式操作系统陆续设计和开发出来，如OSF/1，NeXT，iOS等等。

IBM大型机操作系统—OS/390、z/OS

- 在PC机时代，人们曾经估计大型机会衰亡，但IBM S/390是大型机复活的一个典型。
- 90年代中期，金融、电信、电子商务发展刺激对计算能力的要求，导致大型机市场的再度升温。
- 三十年改进，**IBM System 390**成为具有高可靠性、可扩展性及安全可用性的现代大型机系统。
- *大型机上的主要程序设计语言：COBOL (擅长数据处理)*
- 近十年，IBM的System z系列大型机配置 **z/OS**。
- IBM最新的zEnterprise大型机能够支持 z/OS、IBM AIX、Linux和微软Windows。

Internet时代与Linux

- 1990年秋，20岁的Linus Torvalds在芬兰赫尔辛基大学学习操作系统课程。因为上机需要排队，Linus买了台386PC机，并邮购了MINIX。
- Linus需要访问Usenet新闻组，于是他编写了从调制解调器上接发信息的程序，显示器、键盘和调制解调器的驱动程序，软盘驱动程序，文件系统。
- 至此，拥有了一个OS原型(至少是一个内核)。
- 1991.10.5，Linux 0.02在互联网上“正式”发布。到1994年Linux内核1.0版发布时，已有相当多的操作系统爱好者和网络黑客为Linux贡献了大量代码 。
- 现在，内核的改进(补丁)由其他人完成，Linus的主要任务是对内核的维护和决定是否采用某个补丁程序。

方兴未艾的Linux

- Linux本身并不是UNIX，但在界面和接口标准上与UNIX完全兼容，所以称之为“类UNIX OS”。
- https://www.kernel.org/。
- 开放内核源代码，免费。可修改、定制。安全稳定。支持多种硬件平台，裁剪后还可作为嵌入式操作系统。不兼容Windows软件。
- 基于Linux内核的发行版(Linux Distribution)：红帽，红旗，麒麟，Mint，Ubuntu，Debian，Fedora，openSUSE 等350余个。(不全免费)

无处不在的嵌入式操作系统

- 智能手机、平板电脑等触屏移动设备领域：
  - Android—Google的开源智能手机OS，基于Linux
  - iOS—Apple的移动设备OS
  - Windows Phone，Symbian，Java ME，BlackBerry，Kindle
- 嵌入式Linux：μClinux，MontaVista Linux，KaeilOS，μlinux，Ubuntu Mobile等

作业：

- 复习题：2，3
- 补充习题：

在单CPU和两台I/O设备（I1和I2）的多道程序设计环境下，同时投入3个作业运行。其执行轨迹如下：

Job1: I2(30ms), CPU(10ms), I1(30ms),CPU(10ms), I2(20ms)
Job2: I1(20ms), CPU(20ms), I2(40ms)
Job3: CPU(30ms), I1(20ms), CPU(10ms), I1(10ms)

设CPU, I1和I2都能并行工作，作业优先级从高到低依次为Job1, Job2, Job3，优先级高的作业可以抢占优先级低的作业的CPU，但不可抢占I1和I2。求多道作业并发执行时的：
（1）CPU利用率。
（2）I1、I2的设备利用率。