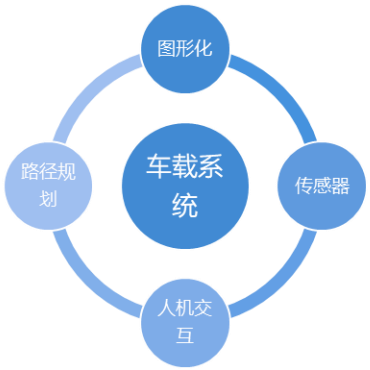
计算机操作系统

教师：王雷

82316284，wanglei@buaa.edu.cn

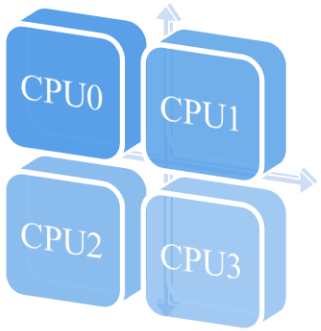
工作简介

- 车载、智能座舱、无人机等混合关键系统嵌入式虚拟化



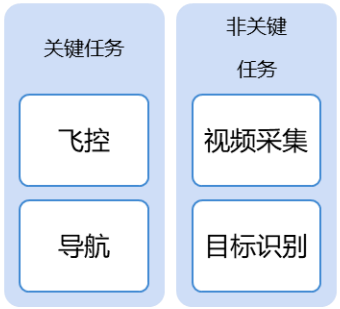
专用->通用

图形化用户界面、传感器数据采集、人机交互、路径规划等



单核->多核

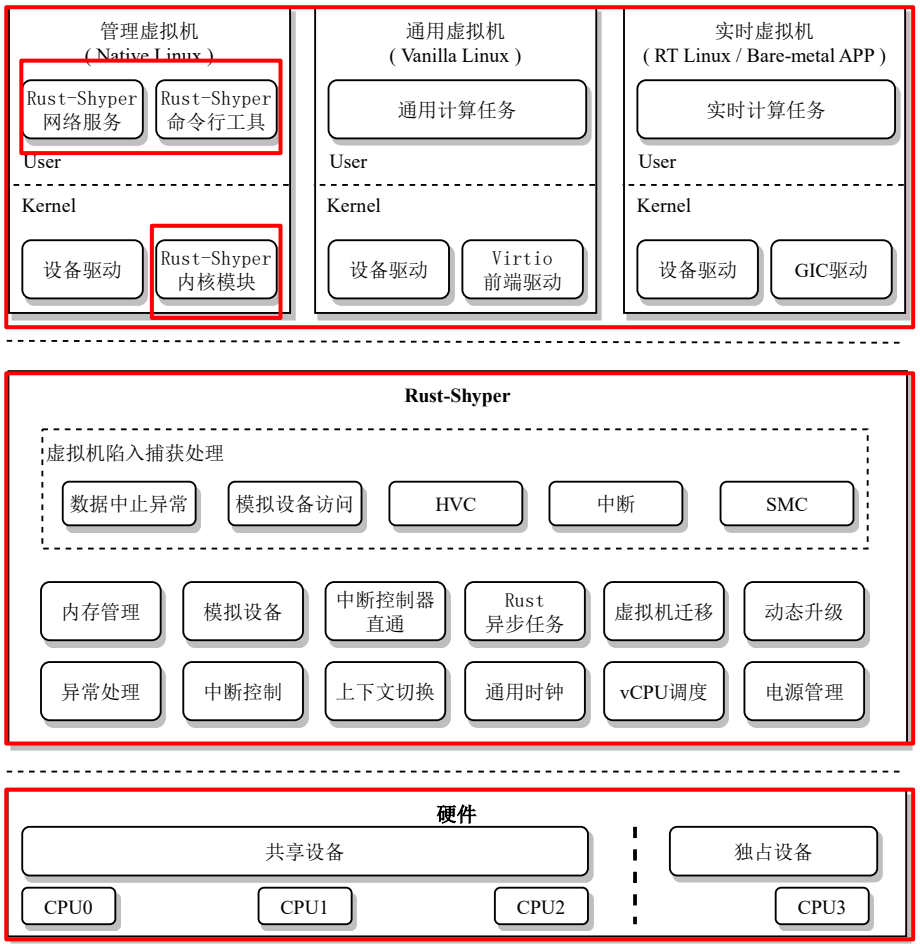
硬件处理器包含多个核心从而可以同时运行多个任务



简单一致->混合关键性

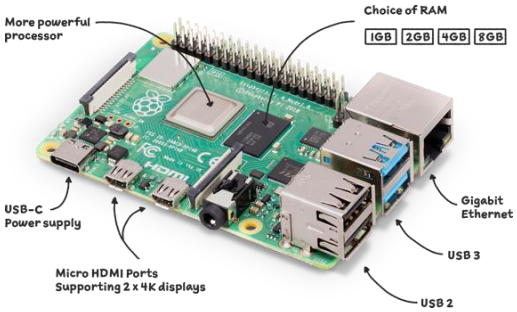
需要同时运行和管理关键任务和非关键任务

Rust-Shyper系统架构设计



- 基于Rust语言的Type I虚拟机监控器，具备更好的隔离性和更轻量的代码实现。基于AArch64、能够支持Linux、裸应用等系统软件的运行
- 灵活、动态创建并配置虚拟机，能够为实时操作系统创建实时虚拟机，保障嵌入式场景的实时需求
- 具备资源隔离属性，保障不同虚拟机间的隔离性
- 实现多种模拟设备，提供基本的磁盘、网络、串口等设备的虚拟化实现，提高资源利用率
- 提供传统虚拟机迁移机制以及本地热更新机制，保障虚拟机监控器正常运行过程中的可靠性

- Rust-Shyper目前支持的平台



树莓派4B

- 支持虚拟机热更新
- 设备直通保证IPA与PA一致且连续

Nvidia TX2

- 主要开发平台
- 支持Smmu-v2
- 支持虚拟机热更新

QEMU

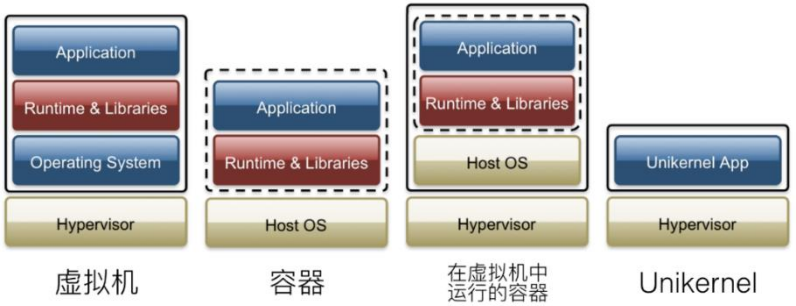
- 需要修改并编译QEMU源代码，来提供更多串口 qemu-system-aarch64之提供一个
- - 嵌套虚拟化，可能存在问题，仅用作验证，不用于功能测试

Unishyper：基于Unikernel的操作系统内核

嵌入式虚拟化与Unikernel

- 虚拟化允许不同的OS以及实时任务作为一个个独立的虚拟机（VM）运行在不同的时间和空间分区，从而同时满足不同关键性任务的不同需要
- 主流嵌入式虚拟化软件
  - ACRN、JailHouse、 **Shyper** 等

Why Unishyper? ——实时性问题



Unikernel是用专门的工具链将库操作系统和应用服务编译并链接成为一个密封的、具有单地址空间的、专门的虚拟机镜像

https://arxiv.org/abs/2104.12721v1

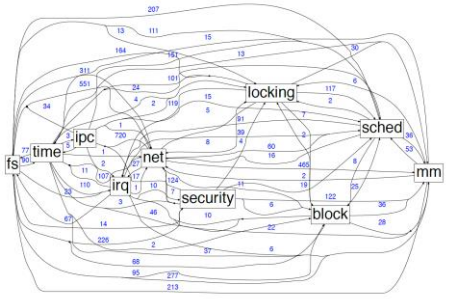


Figure 1. Linux kernel components have strong inter-dependencies, making it difficult to remove or replace them.

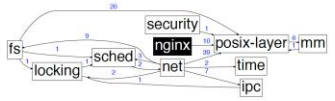


Figure 2. Nginx Unikraft dependency graph

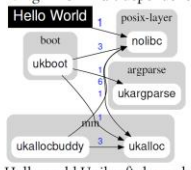
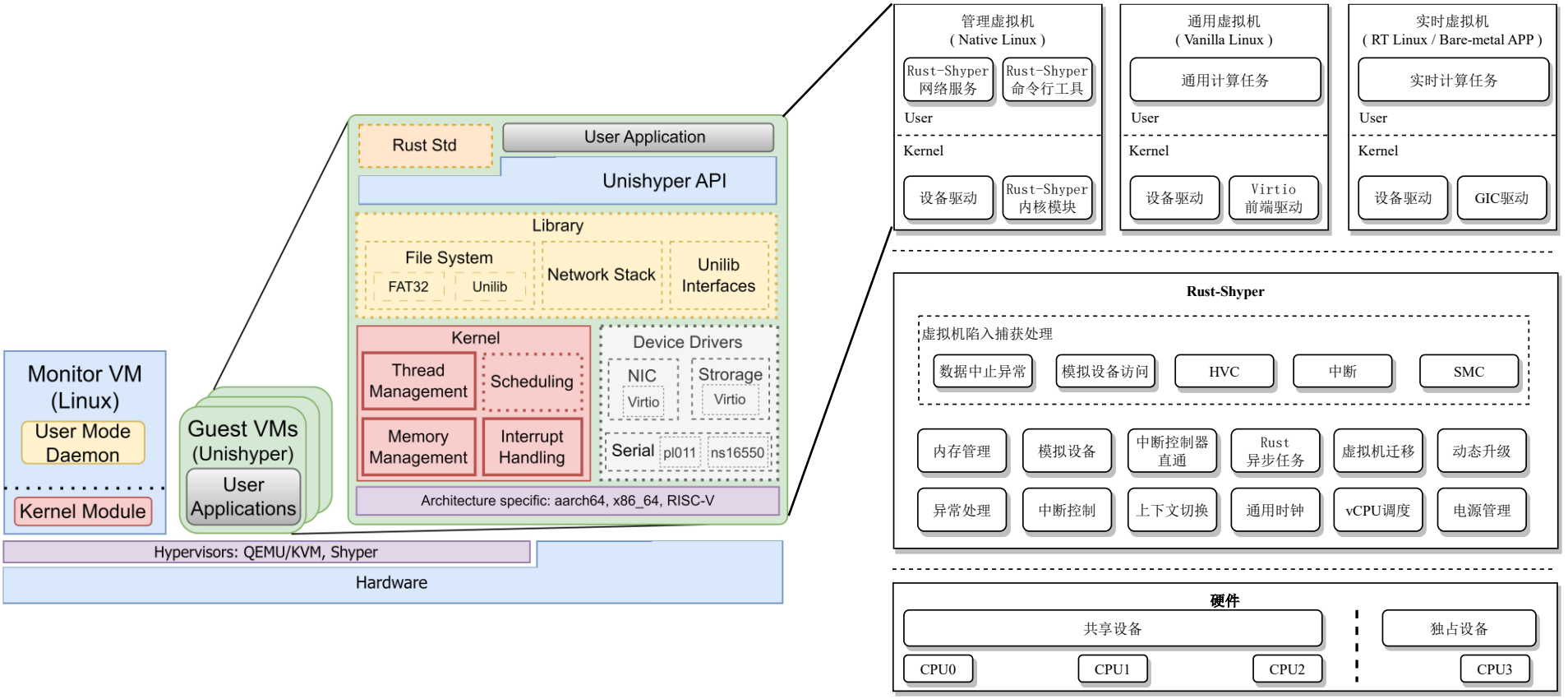


Figure 3. Helloworld Unikraft dependency graph

Unishyper系统设计

- 单地址空间与单特权等级设计、多线程和多平台支持
- 高可靠：硬件支持的单地址空间访存隔离、故障捕捉与恢复机制
- 模块化：丰富的设备驱动与库支持、Unilib 库支持



Unishyper嵌入式多平台支持

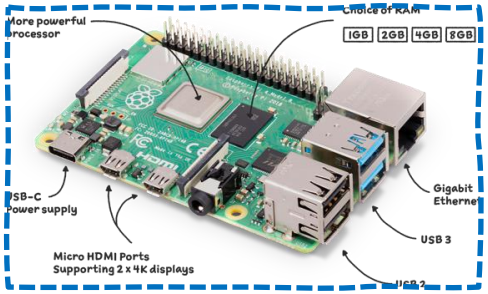
体系结构支持

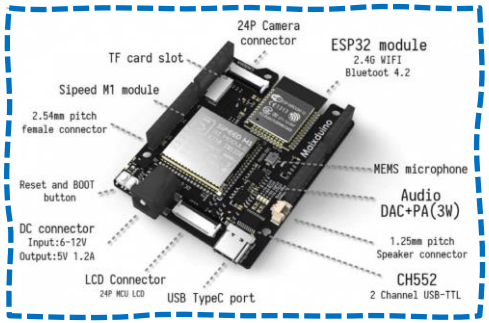
- **Aarch64**
- **x86_64**
- **RISC-V 64**

运行平台支持

- 虚拟化平台
  - **QEMU/KVM**
  - **Shyper Hypervisor**
  - …
- 硬件平台
  - **Nvidia Tegra X2**
  - Raspberry Pi 4 Model B
  - Kendryte K210
  - …

```
Welcome to TX2 Rust-Shyper Hypervisor!
Built At "2023-06-21 12:57:13 CST"
init buddy system
init memory, please waiting...
Memory Heap: base 0x87d53000, size 130 MB / 33453 pages
Memory Heap init ok
Memory VM regions: total 2 region, size 7766 MB / 1988096 pages
Memory VM regions init ok!
Mem init ok
```

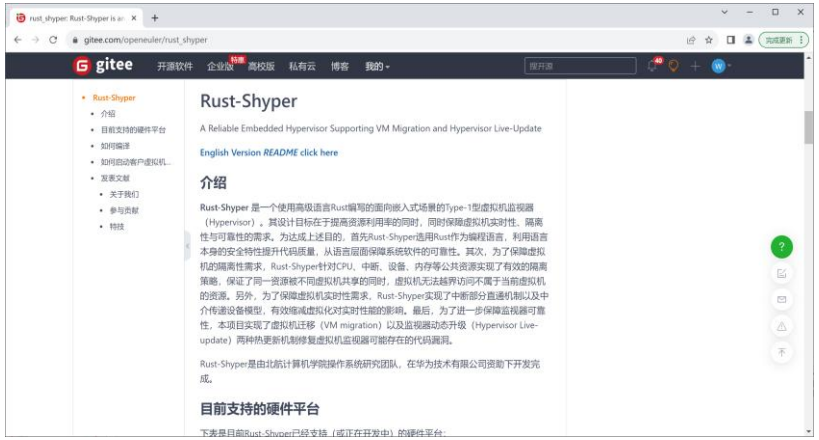


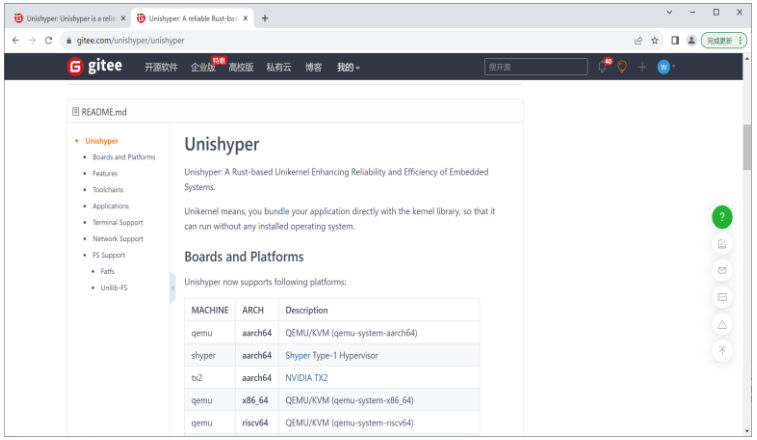



ORBBEC

系统开源情况

- Rust-Shyper已经在openEuler社区开源
- 开源地址：https://gitee.com/openeuler/rust_shyper
- Unishyper已经在gitee上开源
- 开源地址：https://gitee.com/unishyper/





简介

- 理论课
  - 授课方式：在北航在线教学平台中提交作业、讨论答疑；老师会建一个班级微信群进行答疑
  - 考试方式：平时成绩20%+期末考试80%
    - 平时成绩包括作业完成情况、参加讨论、回答问题等，积极参加平时讨论和回答问题最高可加5%
- 实验课
  - 授课方式：在http://os.buaa.edu.cn/发表实验资料，在学校机房介绍实验内容、上机测试和答疑。
  - 考试方式：详细考试与评分标准在http://os.buaa.edu.cn/有详细介绍
- 总成绩：理论课50%+实验课50%

教材

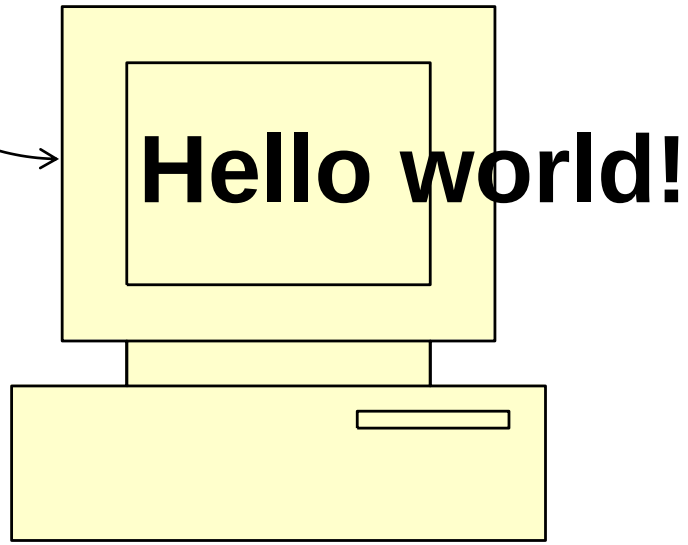
- 任爱华等，操作系统实用教程（第三版），清华大学出版社，2010
- 王雷等，MOS操作系统实验教程，高等教育出版社，2023
- 参考书：
  - Andrew S. Tanenbaum. 现代操作系统 （第三版），机械工业出版社.
  - Abraham Silberschatz. Operating System Concepts (Ninth Edition).
  - Dominic Sweetman. MIPS体系结构透视
  - Randal E. Bryant等. 深入理解计算机系统
  - 陈海波等，现代操作系统原理与实现

内容提要

- WHAT? 什么是操作系统?
- WHY? 为什么学习操作系统?
- HOW? 怎么学习操作系统?

An example

```
#include <stdio.h>
main() {
    printf("hello world!\n");
}
```



- 编译成可执行文件
- 用户告诉shell执行该可执行程序
- 创建一个新的子进程
  - 创建进程控制块
- 装入hello程序
  - 操作系统找到该程序，检查其类型
  - 检查程序首部，找出代码段和数据段的地址
  - 可执行文件映射到进程结构
  - 设置CPU上下文环境，并设置程序开始处
  - 调度hello程序
- 执行程序的第一条指令
  - 执行失败，缺页中断发生
  - 分配一页内存，并将代码从磁盘读入，继续执行
  - 更多的缺页中断，读入更多的页面

- printf
  - 操作系统检查字符串的位置是否正确
  - 操作系统找到字符串被送往的设备
  - 设备是一个伪终端，由一个进程控制
  - 操作系统将字符串送给该进程
  - 该进程告诉窗口系统它要显示字符串
  - 窗口系统确定这是一个合法操作，然后将字符串转换成像素
  - 窗口系统将像素写入存储映像区
- 视频硬件将像素表示转换成一组模拟信号控制显示器（重画屏幕）
- 显示器发射电子束
- 你在屏幕上看到hello world!

操作系统的工作

- 程序的执行
  - 负责启动每个程序，以及结束程序的工作
- 完成与硬件有关的工作
- 完成与应用无关的工作
  - 易于使用，基本服务，统一性
- 计算机系统的效率与安全问题

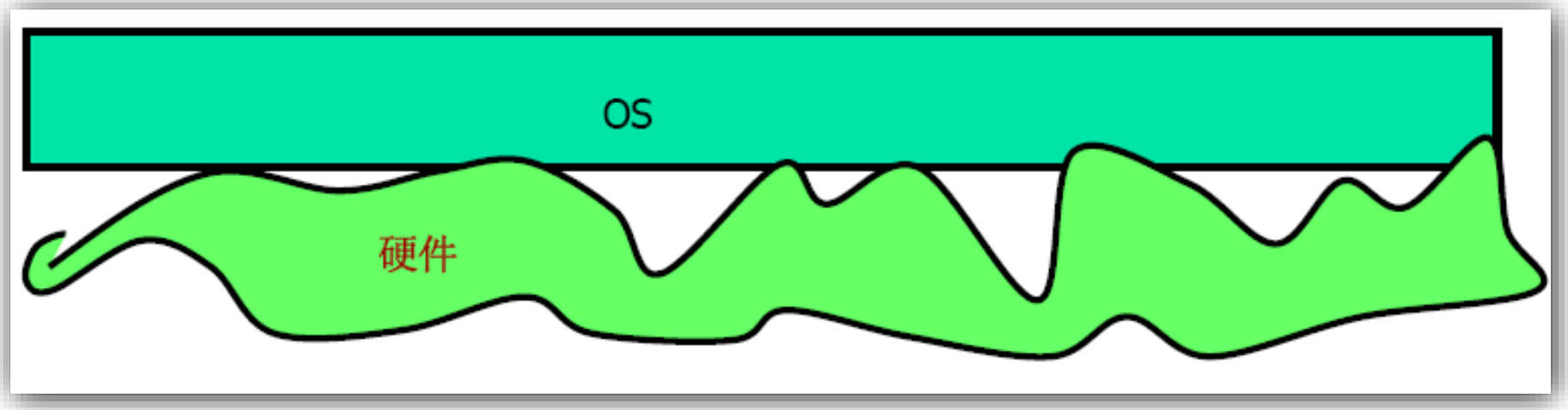
问题1

- 假如没有操作系统，怎样控制硬件?

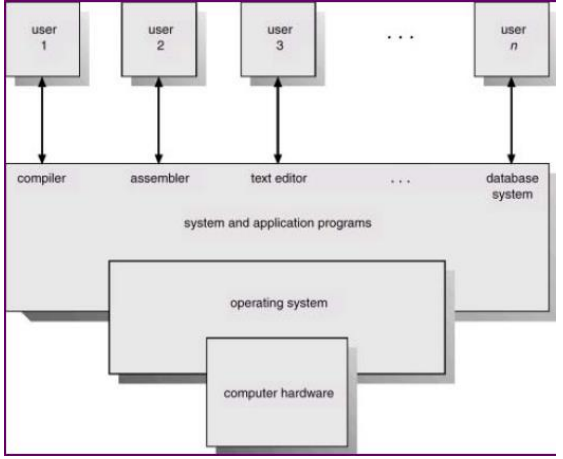
例子：软盘I/O操作

- 控制芯片NEC PD765有16条命令
- 每一条命令向一个设备寄存器装入长度从1 到9 字节的特定数据（读写数据、移动磁头臂、格式化磁道，及初始化、检测状态、复位、校准控制器及设备等）
- 以READ为例：13个参数
- 要读取的磁盘块地址、磁道的扇区数、物理介质的记录格式、扇区间隙、对已删除数据地址标识的处理方法
- 操作结束时，控制器芯片在7个字节中返回23个状态及出错字段
- 软盘程序员还要保持注意步进电机的开关状态

- 抽象！
  - 管理硬件、方便使用、提供保护

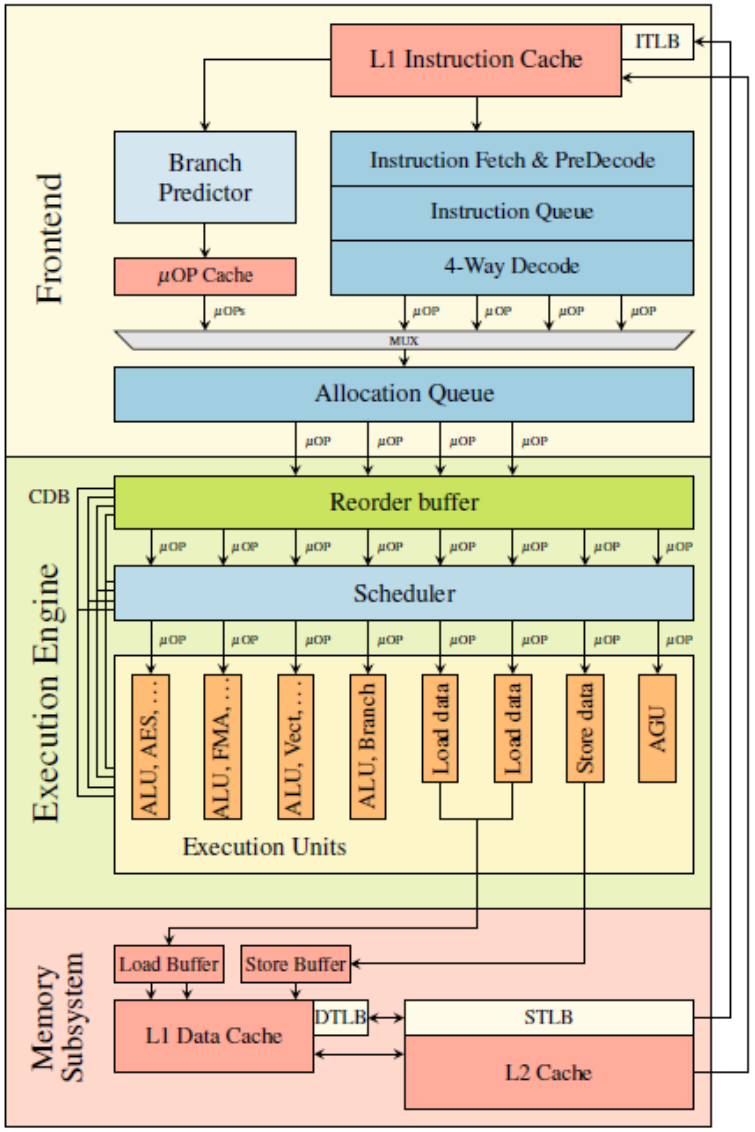


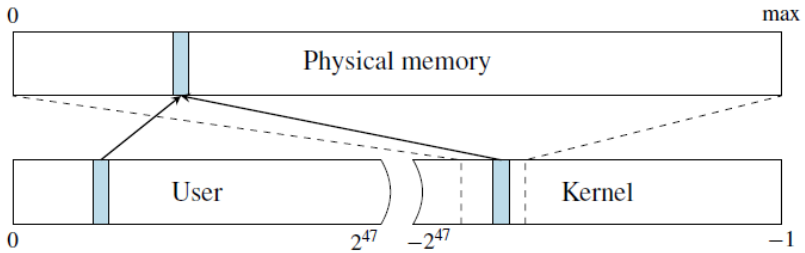
学习操作系统课的重要性



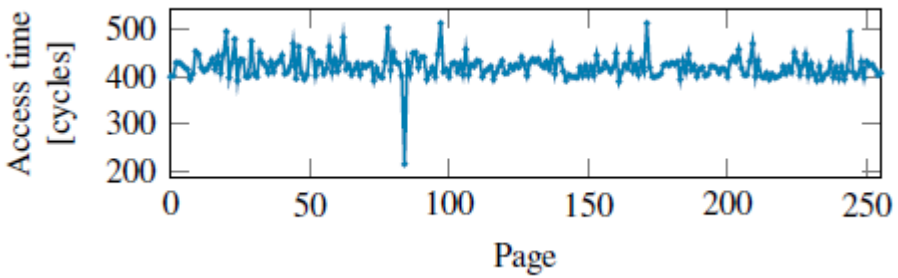
- 国家战略需求“缺芯少魂”
  - 卡脖子问题（排第三位）
  - 美国对华为的制裁
  - 微软垄断及后门问题
  - 2018年10月起 Google正式对欧盟区域的 Android进行收费
- 使用计算机必须要使用操作系统
  - 交互式、编程（系统调用）
- 操作系统的设计原理体现了软硬件最新的发展
  - 计算机体系结构（巨型机、大型机、微机、工作站、多核、XPU等）
  - 程序设计方法（并发、面向对象、结构化）
  - 加深对使用的OS的理解，有利于深入编程
  - 编程时借鉴操作系统的设计思想和算法
  - 大型系统的设计
- 将来个人的发展

Meltdown & Spectre



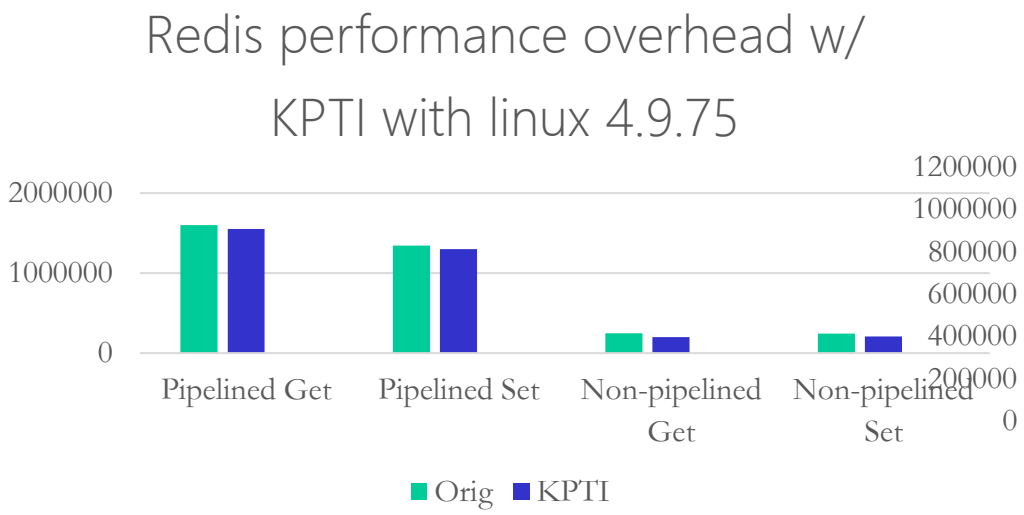


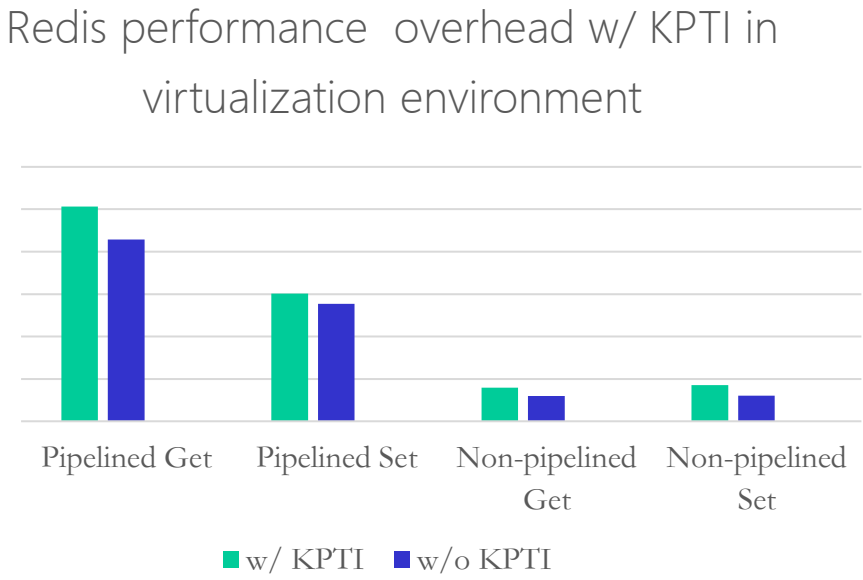

```
1 ; rcx = kernel address
2 ; rbx = probe array
3 retry:
4 mov al, byte [rcx]
5 shl rax, 0xc
6 jz retry
7 mov rbx, qword [rbx + rax]
```



KPTI

- Kernel Page Table Isolation
- 内核与应用使用不同的页表
  - 用户态无法直接寻址内核空间的数据
  - 缺点：切换页表需要刷新TLB，影响性能
    - Tagged TLB可避免切换时刷新，但Linux没有用



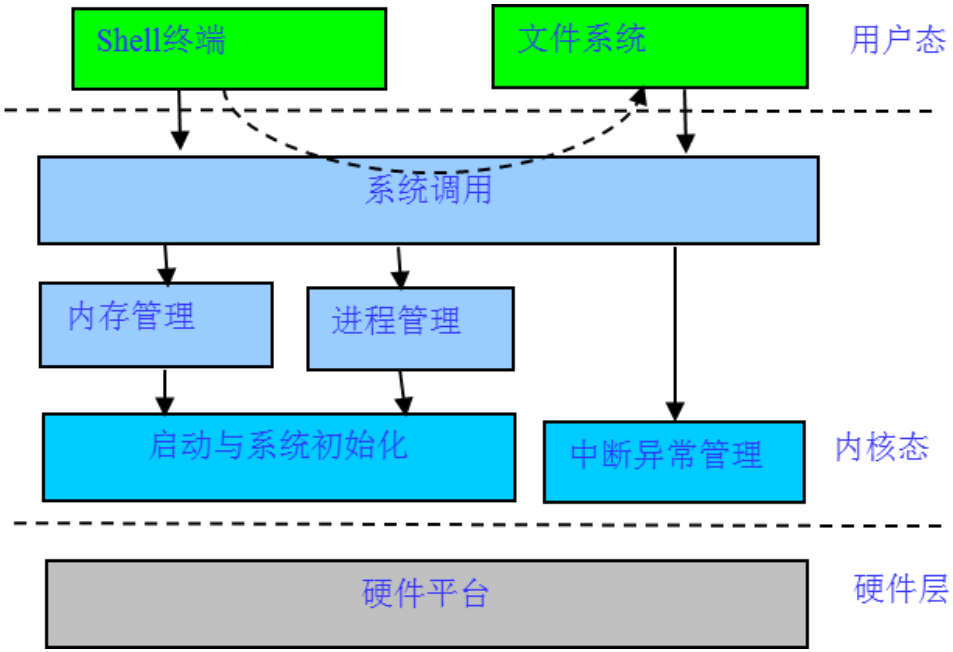


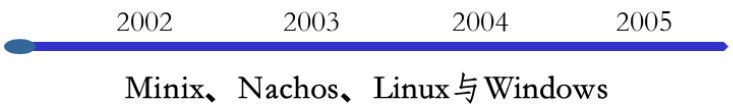
课程目标

- 掌握操作系统的基本原理
  - 进程管理、存储管理、设备管理和文件系统
- 对操作系统的基本概念、基本结构及运行环境有深入的认识
  - 具有分析操作系统的能力
- 系统级编程能力的提高
  - 系统调用的使用、内核开发
- 深入到操作系统内部，理解并掌握操作系统的基本原理、设计方法和实现技术
  - 操作系统设计、并发程序的设计方法
- 思维方式的学习：权衡（Trade Off）、抽象
- 了解操作系统的演化过程、发展研究动向、新技术以及新思想

学习方法

- 按时上课，认真听讲，理解原理
- 阅读参考书，扩展视野
- 阅读代码，分析实例
- Lab0-lab7上机实践
- 记笔记，思考，讨论，提问
- Learn OS concepts by coding them!




2002
2003
2004
2005
Minix、Nachos、Linux与Windows


“系统能力”教改
目标：本科生开发一个小型“操作系统”
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
教学设计
调研
MIPS移植
整体式实验
增量式实验
增量式+挑战性
增量式+挑战性+竞争性
实验环境开发
MIPS单机
在线虚拟机
云平台
学习行为量化分析
问题发现与改进
教学效果预测
个性化推荐

操作系统引论

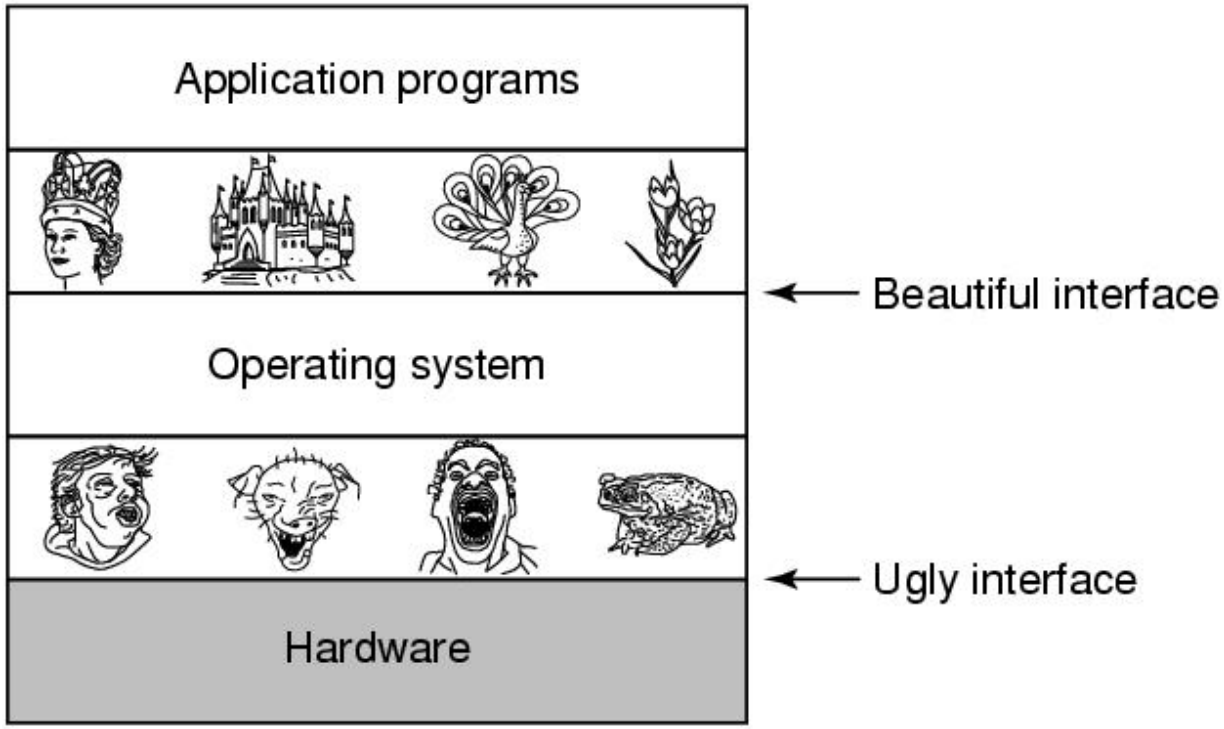
- 什么是操作系统
- 操作系统简史

操作系统定义

- 操作系统是一组管理计算机硬件资源的软件集合，它向计算机程序提供共性的服务
  - 提供一个计算机用户与计算机硬件系统之间的接口，使计算机系统更易于使用。（使用者）
  - 有效地控制和管理计算机系统中的各种硬件和软件资源，使之得到更有效的利用。（资源管理者）
  - 合理地组织计算机系统的工作流程，以改善系统性能（如响应时间、系统吞吐量）。

自顶向下看操作系统

- 如何管理纷繁复杂的硬件资源?
- 屏蔽复杂性：抽象
  - 创建抽象
  - 实现和管理抽象
- 例如：文件系统



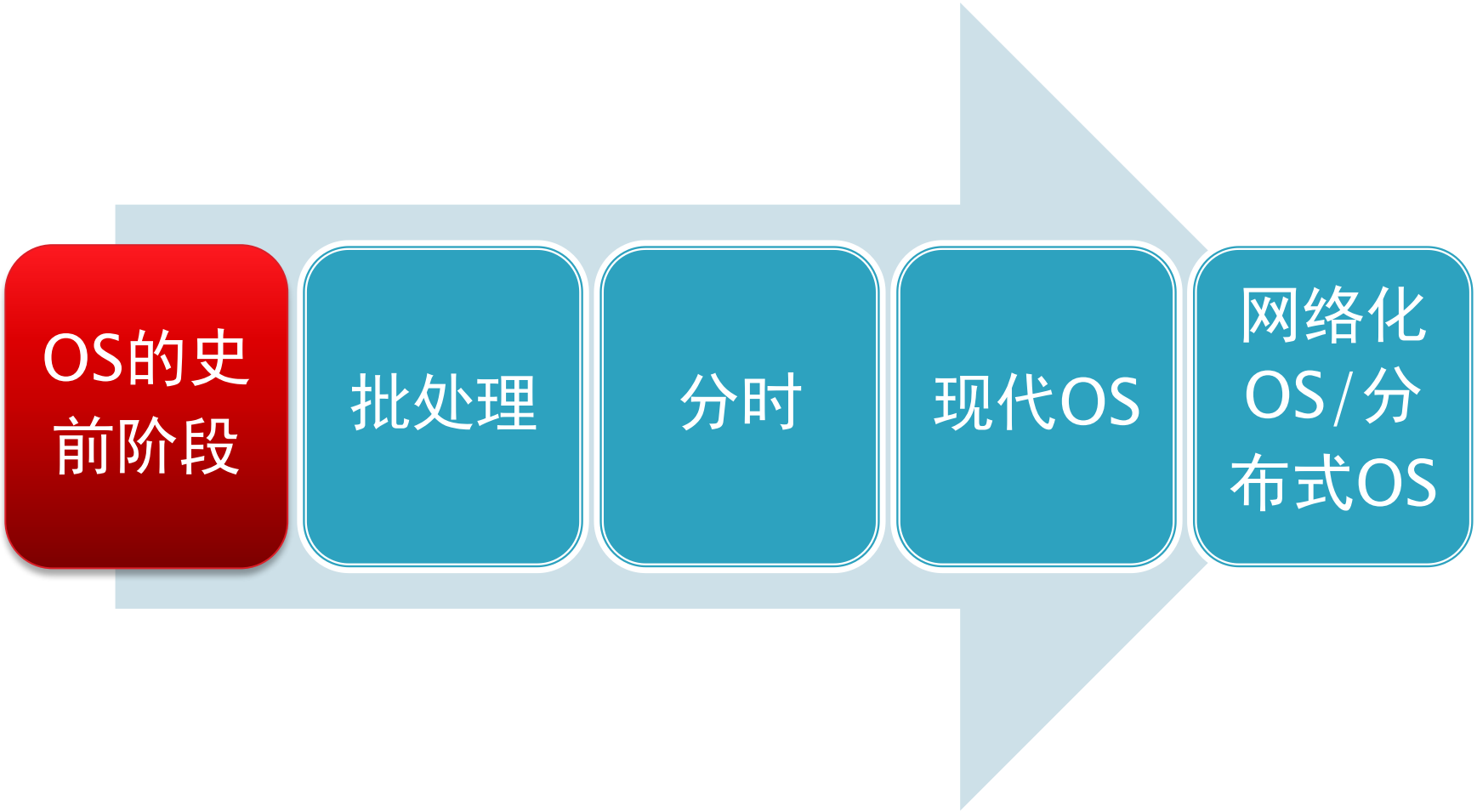
自底向上看操作系统

- 如何向上层应用提供服务?
  - 支持多个用户、多个程序
  - 资源竞争
- 资源的复用（共享）
  - 时间维度
  - 空间维度
- 例如：打印机

操作系统引论

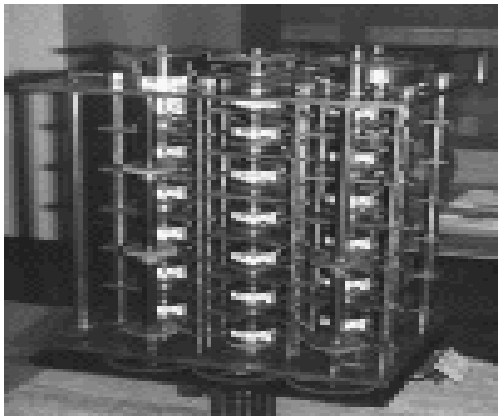
- 什么是操作系统
- 操作系统简史

OS的历史沿革



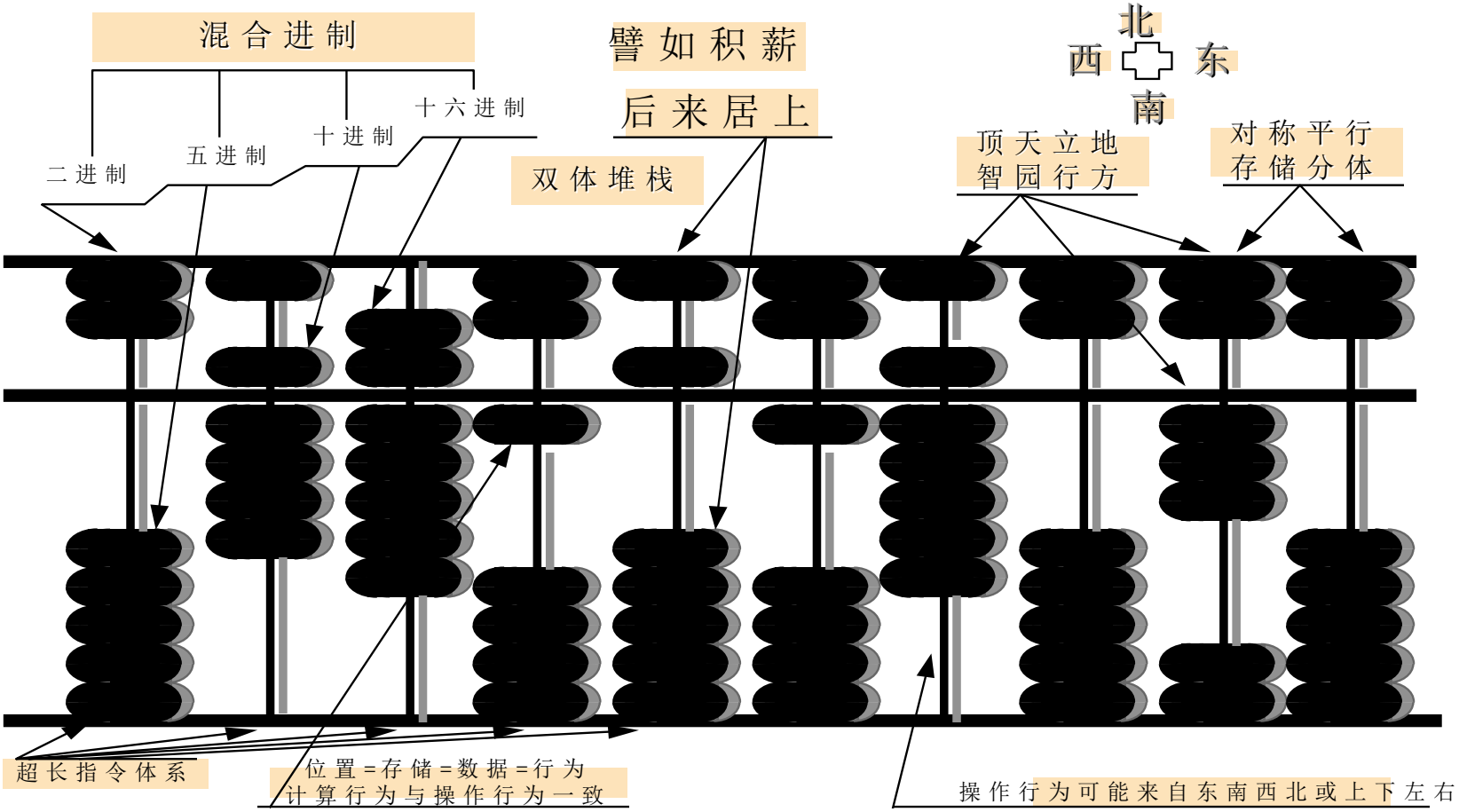
史前的计算

- 算盘
- Pascal：*1642年 法国数学家 十进制加法器 （世上第一台真正的机械计算机）。*
- Charles Babbage：*1822年 英国数学家 研制成差分机模型，研制分析机没成功。*
- Zuse：*1941年 德国工程师 完成继电器式通用计算机Z-3。*



Zuse Z3 replica on display at Deutsches Museum in Munich

算盘



算盘由珠、棍、架组成（硬件），口诀（软件）及操作反映了运算过程及结果

现代计算机奠基人

- Turing：*1936年一篇关于判定性问题的论文，提出了图灵机。ACE(Automatic Computing Engine).*
- *冯·诺依曼：1945年3月，Von Neumann领导的小组发表了二进制的程序存储式的电子数字自动计算机EDVAC方案，1945年7月，Von Neumann等人又提出更为完善的的设计报告，宣告了现代计算机结构思想的诞生。1951年EDVAC完成。*
- J. Presper Eckert and William Mauchley

操作系统简史

- 1946~1955年 电子管
- 1955~1965年 晶体管 & 监控系统
- 1965~1980年 集成电路 & 多道程序设计
- 1980~1990年 PC机 & 微机操作系统
- 1990~ 年 分布式与嵌入式系统

1946~1955年 电子管

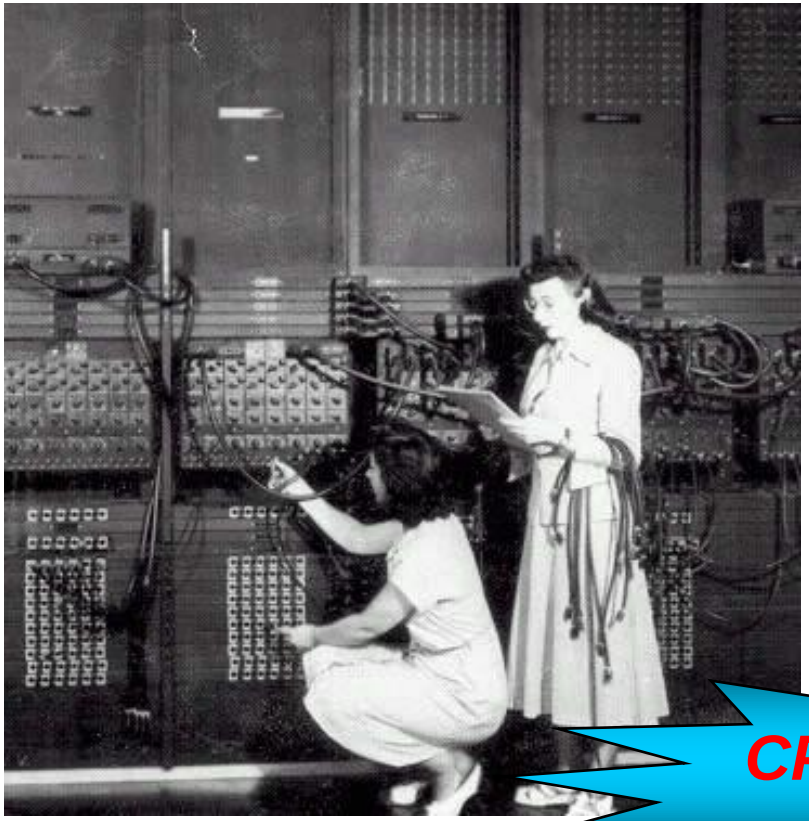
- 1946年2月 美国宾夕法尼亚大学莫尔学院制成世界上第一台数字计算机ENIAC（未采用二进制操作和存储程序控制、未具备现代电子计算机的主要特征）(重30吨、占地170平方米、18000个电子管、5000次/秒)
- 1949年英国剑桥大学的M.V.Wilkes在EDVAC方案的启发下研制成世界上第一台程序存储式的现代计算机EDSAC。（变址、宏指令、微程序、Cache.）

1952年

IBM 701，“国防计算器”，月租金15000美元；
无OS，每个软件的程序员都要从头开始编写程序。

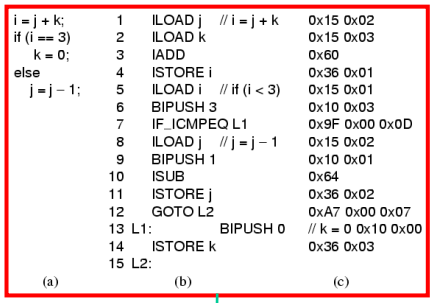
计算机软件与硬件：“绑定”到“分离”

- 面向特定计算机，采用机器/汇编语言实现



ENIAC上的软件：手工修改逻辑单元的连接

- 冯.诺依曼：存储程序+条件控制转移
- 香农：二进制演算
- 硬件：晶体管的发明



编译器

晶体管计算机

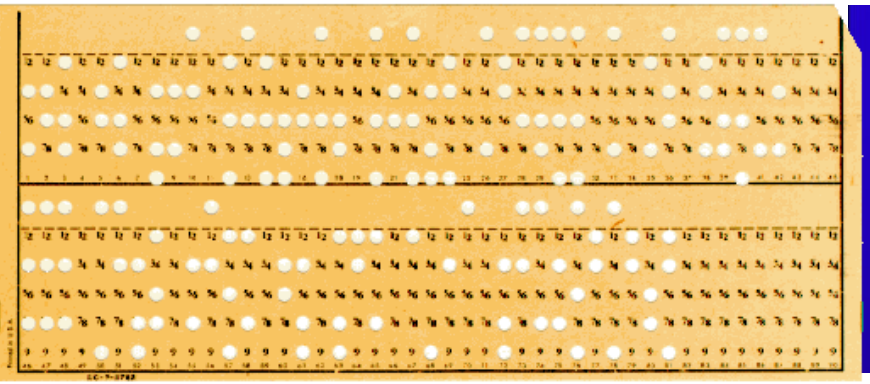
CPU与人的矛盾

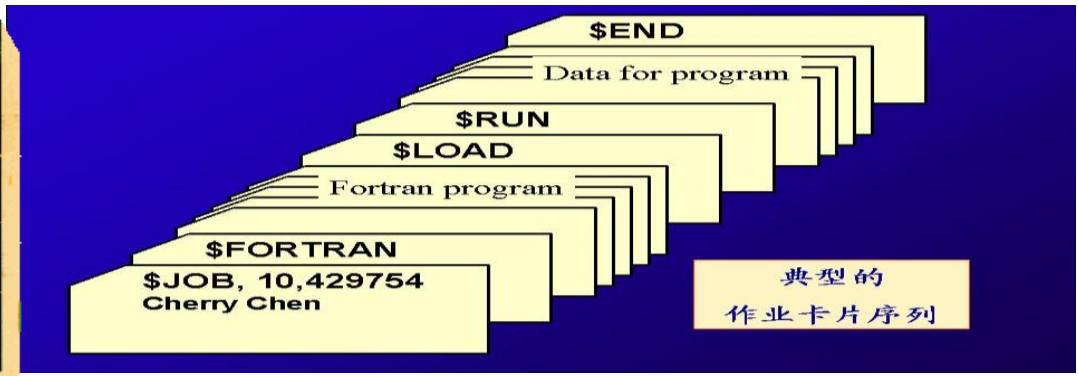
操作系统的名称

- 监控系统（Monitor)
- 执行系统（程序）(Executive System (program))
- 控制系统（程序）(Control System program))
- 管理程序(Supervisor, Supervisory System)
- 核心程序(Kernel)
- 操作系统(Operating System)

1955~1965年 晶体管 & 监控系统

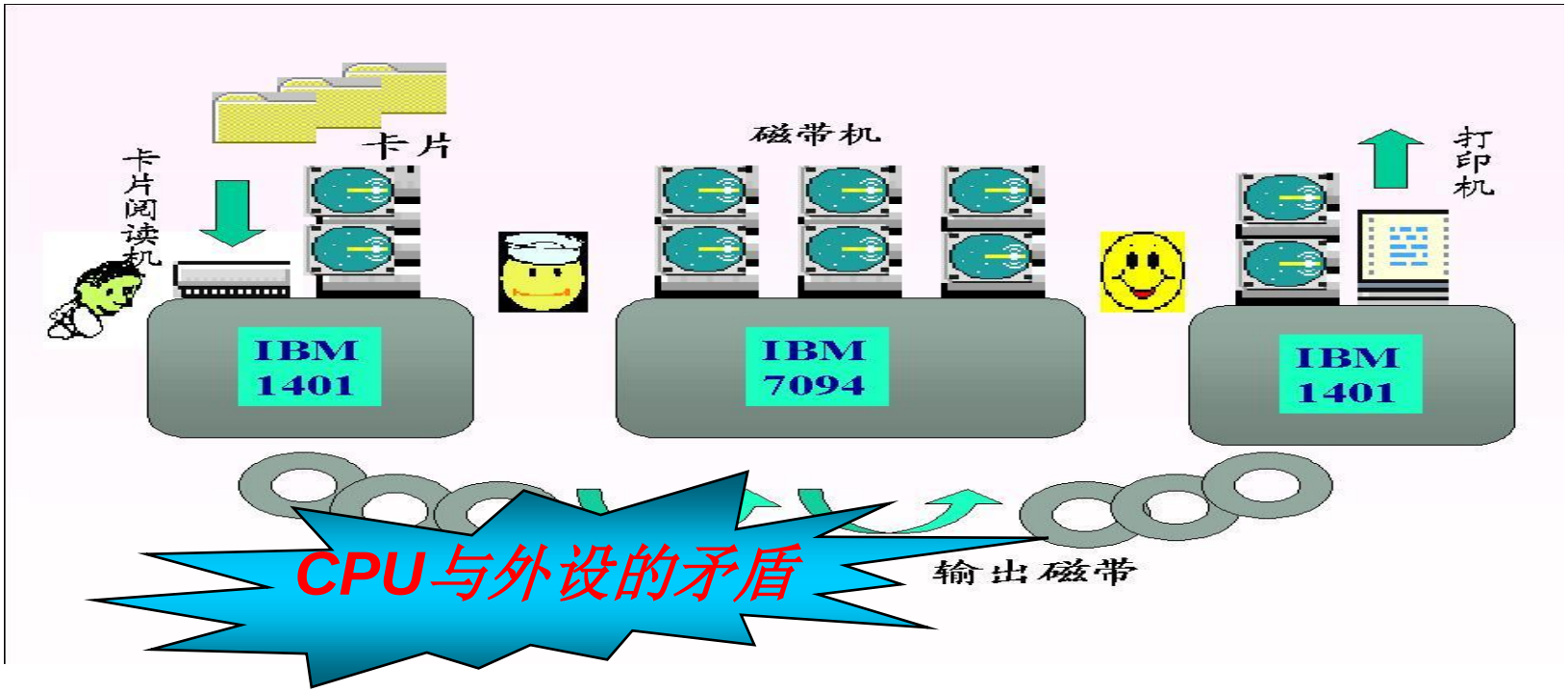
- IBM的1401（卫星机）和7094（主机）。
- 第一个批处理操作系统GM-NAA I/O
- FORTRAN语言出现（语言的历史70年代FORTRAN为主，80年代C语言为主，90年代C++为主，2000是JAVA，2010年？？）。
- 操作系统的代表为FMS (the Fortran Monitor System)，批处理系统。





批处理技术

- 批处理技术：计算机系统对一批作业进行处理的技术。



1965~1970年 小规模集成电路 & 多道程序设计

- OS/360开发失败，导致了软件工程的诞生，从另一个意义上说，它也有成功之处，改变了上一代操作系统的问题，提出了多道程序设计（进程）、向下兼容的思想。商业上成功。
- 分时系统，CTSS、MULTICS、UNIX。

多道批处理系统

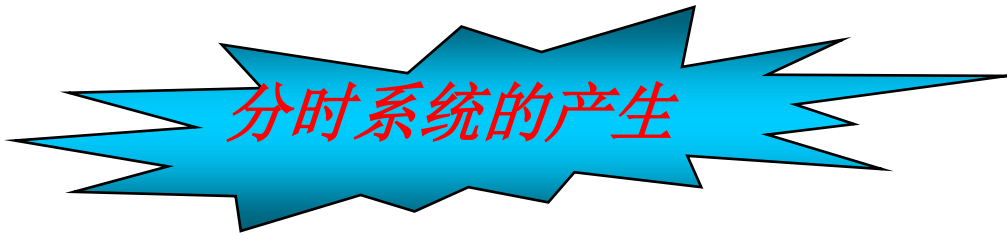
- 20世纪60年代中期，在前述的批处理系统中，引入多道程序设计技术后形成多道批处理系统（简称批处理系统），具有两个特点：

  ① 多道：系统内可同时容纳多个作业。这些作业放在外存中，组成一个后备队列，系统按一定的调度原则每次从后备作业队列中选取一个或多个作业进入内存运行，运行作业结束、退出运行和后备作业进入运行均由系统自动实现，从而在系统中形成一个自动转接的、连续的作业流。

  ② 成批：在系统运行过程中，不允许用户与其作业发生交互作用，即：作业一旦进入系统，用户就不能直接干预其作业的运行。

多道批处理系统

- 优点：
  - 系统吞吐量大
  - 资源利用率高
- 缺点：
  - 平均周转时间长
  - 不能提供交互作用能力



操作系统的发展 – Time Sharing

- What 分时系统
  - 将CPU处理时间分割为多个时间片，将时间片分给不同程序，达到多个程序“同时”运行的效果
- Why 分时系统
  - 批处理系统一次执行一个程序，I/O过程CPU空转
  - 为进一步提高CPU利用率，支持多用户、多进程
- 出现时期(1957-~)
  - 分时系统的概念John Backus（1954）、Bob Bemer（1957）提出
  - 第一款分时系统CTSS(Compatible)在1961发布
- 两种典型的分时系统
  - 1. Multics/Unix (1968/1970)
  - 2. IBM VM 360/370 (1966/1972)

1983年图灵奖

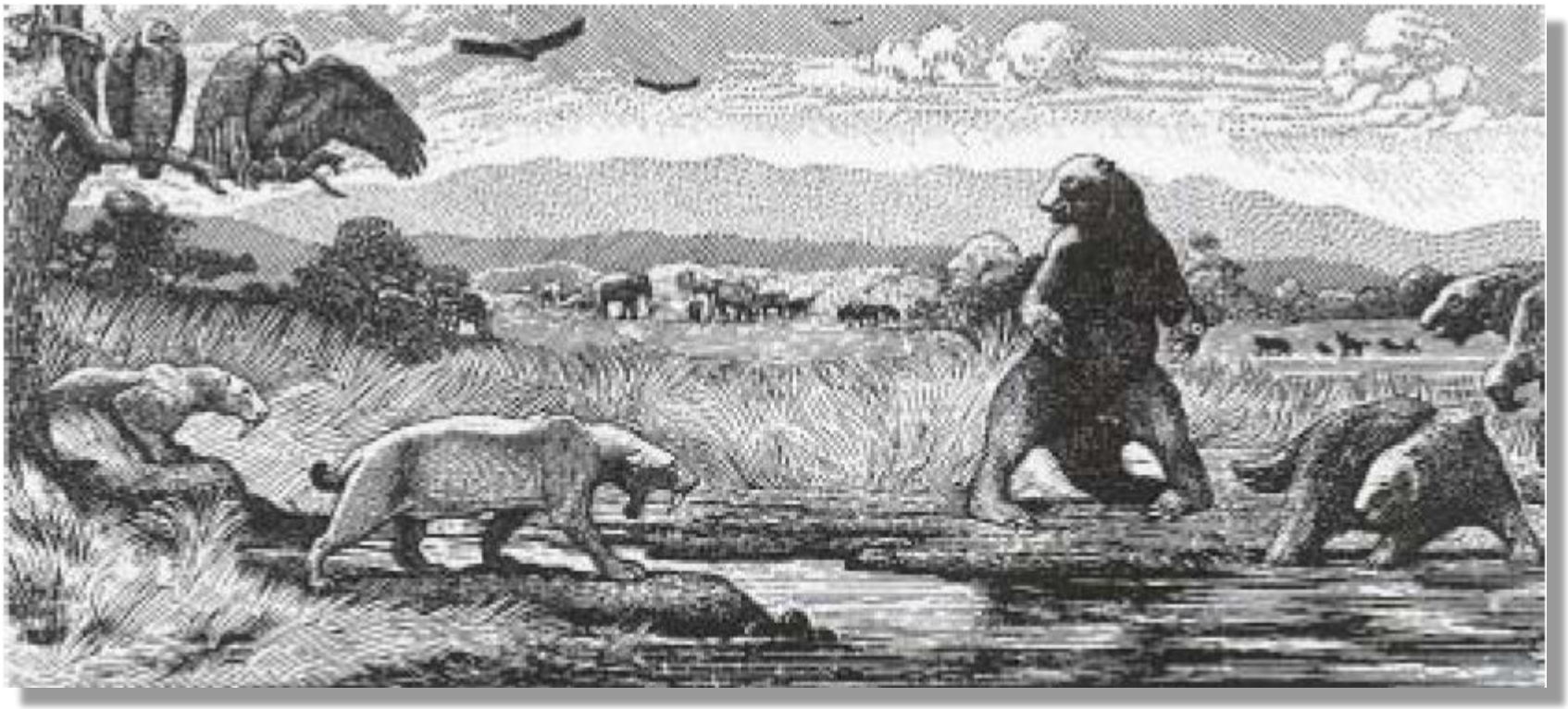
- 1983年Dennis Ritchie和Ken Thompson一同被授予图灵奖
- 以表彰其在通用操作系统理论领域的贡献，特别是Unix操作系统的开发与实现。C语言的发明人。

System/360

- 需求
  - 多数厂商有几条完全不同的生产线，生产不同的计算机，开发和维护完全不同的产品，对厂商来说是昂贵的。
  - 新用户在开始时只需要一台小计算机，后来可能需要一台大的计算机，而且希望能在新计算机上执行原有的程序。这样，厂家和用户需要软件在不同型号的计算机之间兼容。
- 1964 年IBM 宣布推出System/360计算机系统
  - 是第一个采用小规模集成电路的主流机型
  - 所有的计算机都有相同的体系结构和指令集
  - 在理论上，为一型号编写的程序可以在其他型号机器上运行

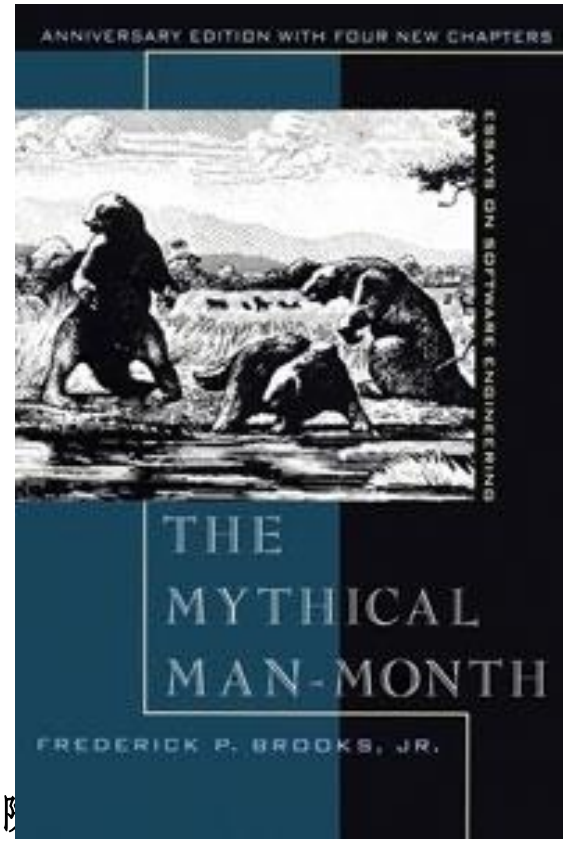
陷入泥潭的巨兽

- IBM无法写出同时满足互冲突需求的操作系统，其实别人也一样不能完成这项工作任务
- 数千名程序员写的数百万行汇编语言代码，系统自身占据了大量存储空间和一半的CPU时间
- 数百万行汇编代码中有成千上万处错误，IBM不断发行新的版本试图更正这些错误
- 每个新版本在更正老错误的同时又引入新错误，所以随着时间的流逝，错误的数量大致保持不变
- IBM 的 OS/360 发布时，带着已知的 1000 个错误



Frederick Brooks

- 因在计算机体系结构、操作系统和软件工程方面的贡献而获奖。
- IBM OS360的总设计师。
- 软件工程中的瀑布模型的提出者。



1980~1990年 PC机 & 微机操作系统

- PC机
- MSDOS、WINDOWS 95、WINDOWS NT、WINDOWS 2000
- 类UNIX(Linux, Freebsd等)。


Great Wall
Great Wall

	1981	2006	Factor
CPU MHz, Cycles/inst	10 3—10	3200x4 0.25—0.5	1,280 6—40
DRAM capacity	128KB	4GB	32,768
Disk capacity	10MB	1TB	100,000
Net bandwidth	9600 b/s	1 Gb/s	110,000
# addr bits	16	32	2
#users/machine	10s	≤ 1	≤ 0.1
Price	$25,000	$4,000	0.2

1990~　　年 分布式与嵌入式系统

- Cluster of Workstation, Network of Workstation, Grid, Cloud
- 实时操作系统：Psos, VRTX, RTLinux, VxWorks
- 智能操作系统：智能车载、无人机载、智能家庭
- AI + IoT + 5G + Cloud + …

5G

10-100X ↑吞吐量
10-100X ↓时延

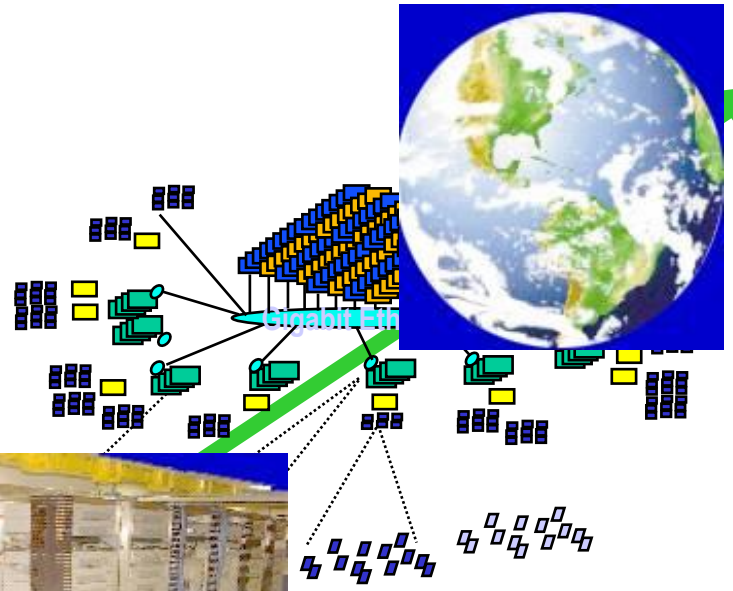
AI 硬件

100X 计算能力

IoT

100X 设备数

- 万物互联的基础——操作系统



Scalable, Reliable, Secure Services
可伸缩、可靠、安全的服务

Mobile, Ubiquitous Systems
移动、普适计算系统

MEMS for Sensor Nets
面向传感器网的微机电系统

问题2

- **哪些因素导致了批处理系统、多道程序、分时系统的产生?**

概念理解练习题

1. 假如没有操作系统，怎样控制硬件?
2. 什么是多道程序设计？多道程序设计与分时系统的区别是什么?
3. 什么原因推动了操作系统从批处理发展到多道程序，进而发展到分时系统?

- "The C Programming Language", K&R
- "Advanced Programming in the UNIX Environment", W. Richard Stevens
- "Life with UNIX(R) -- A Guide for Everyone”, Don Libes, Sandy Ressler
- “Unix Programming Environment”, Brian W. Kernighan, Rob Pike
- “The Design of the UNIX Operating System”, Maurice Bach
- “The Design and Implementation of the 4.4 BSD Operating System”, McKusick等

致谢

- 本讲义内容部分参考了：
  - Andrew S. Tanenbaum. 现代操作系统
  - Abraham Silberschatz. Operating System Concepts
  - 北京大学操作系统课程讲义
  - 清华大学操作系统课程讲义
  - 上海交通大学操作系统课程讲义