

深入理解操作系统

宜宾学院教学试点
宫晓利 南开大学

宫晓利 副教授，博士生导师



个人简介

九三学社南开大学 副主委

南开大学物联网工程系副主任，博士生导师

南开大学系统与网络研究所 副所长

脑科学与智能康复天津市重点实验室 副主任

中国计算机学会体系结构专委会 常务委员

天津市优秀科技特派员

九三学社天津市科技委委员

加州大学洛杉矶分校 访问学者

主持国家重点研发计划、国家自然科学基金

天津市自然科学基金等项目

获天津市科技进步二等奖 4次

什么是操作系统？

这似乎是一个不用回答的问题



操作系统的作用是什么？

- 提供人机交互的图形界面？
- 实现开关机、电话、短信、信号强度、电池电量、应用程序安装？
- 时不时的提供一些发布会的新话题？
- 让未过时的硬件变慢，增加用户的消费意愿。。。。。

几个值得思考的小问题

- 每一部手机的硬件是一样的吗？
 - CPU、内存、屏幕大小。。。。
- 每一个手机上安装的程序是一样的吗？
 - 抖音、微信、B站是否为每一款手机开发了一个app？
 - 当你自己设计了一个新的手机，是不是需要联系每个厂商来写程序？
- 为什么电脑增加了内存之后，程序不用修改就能利用那些新增的内存？
- 为什么程序可以不用考虑硬件的差异？
- 这是如何实现的？



```
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    cout<<"hello world";
    return 0;
}
```

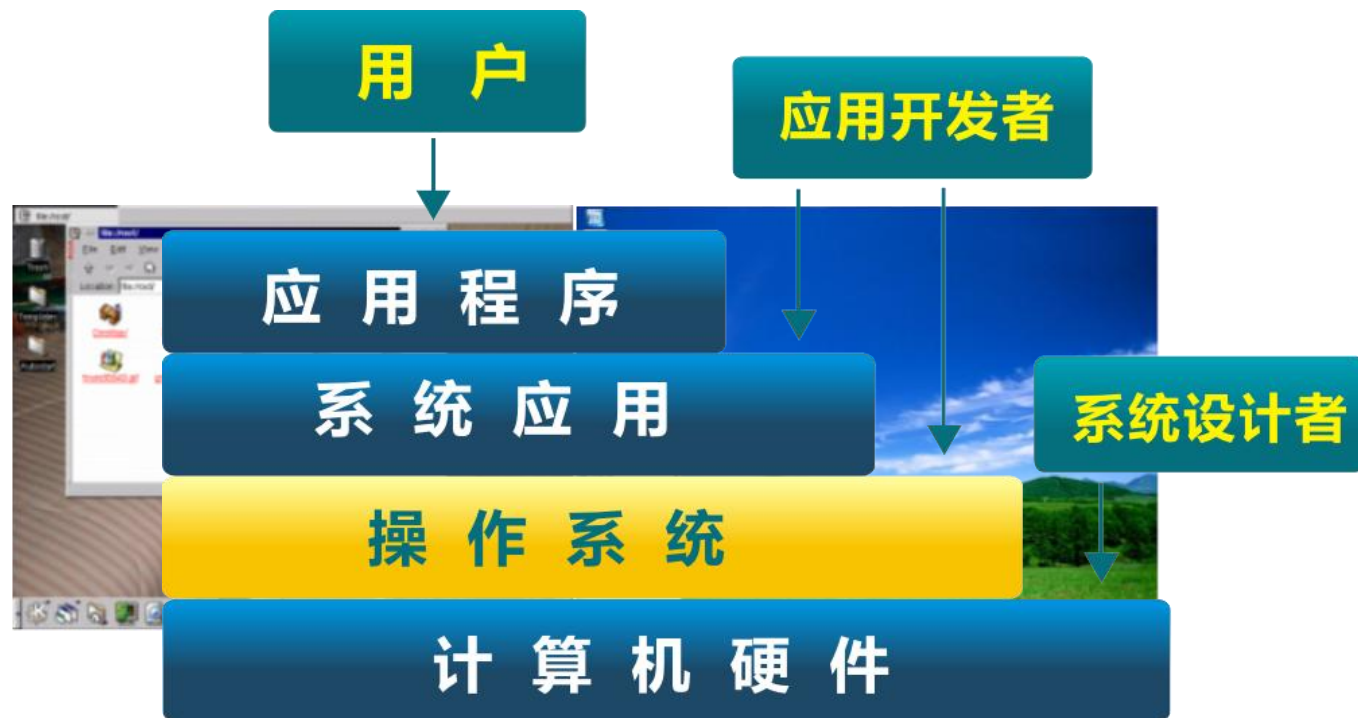

写出hello word发生了什么

- VSCode和操作系统交互
 - 敲击键盘，得到文字
 - 编译、链接，转换成指令
- 你的程序和系统的交互
 - 开始运行，控制屏幕
 - 结束运行
 - 被强制杀死

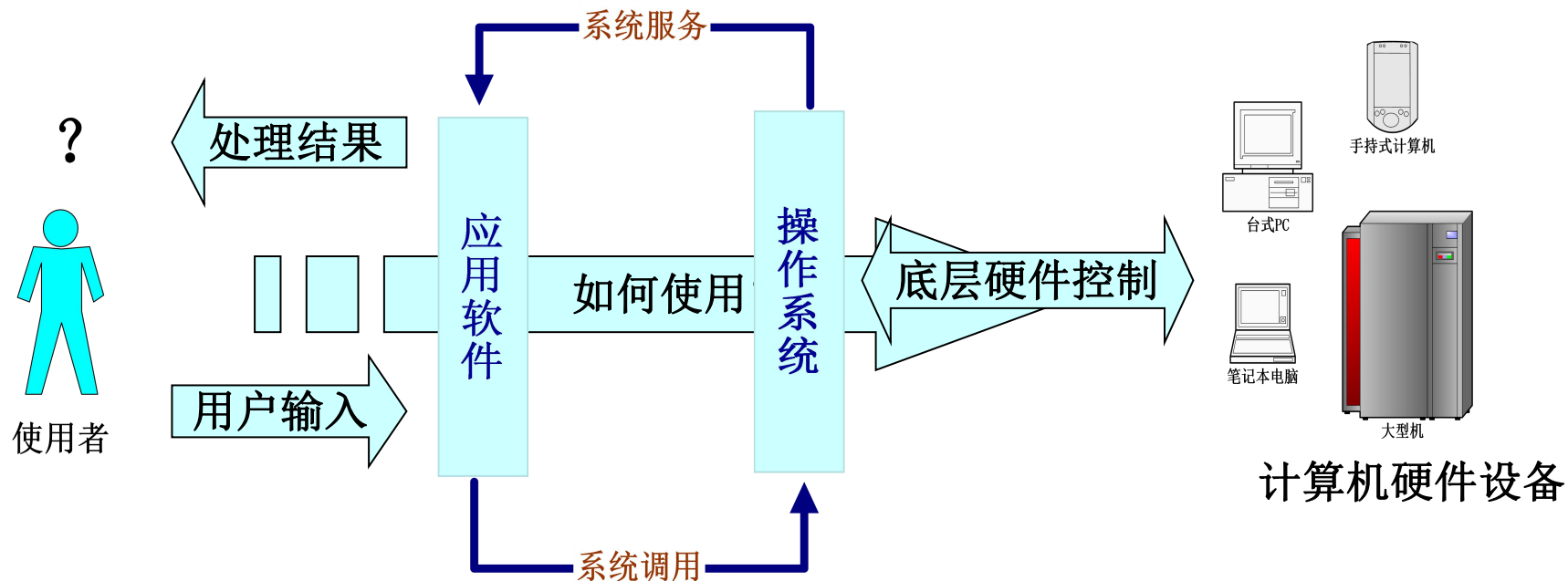
对操作系统的直观感受

- 操作系统是一些函数的集合，应用程序开发人员调用它们，可以减少对硬件细节适配工作
- 操作系统是一个权限更高的软件，它掌管着其他软件的生命周期

操作系统的地位



操作系统在软件中的地位



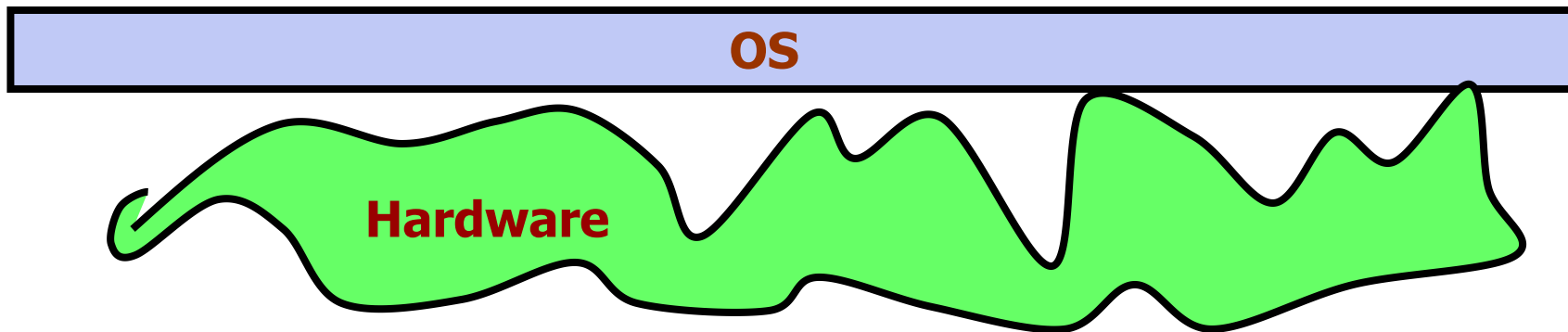
操作系统定义

- 没有公认的精确定义
- 操作系统是一个控制程序
 - ▶ 一个软件
 - ▶ 控制程序执行过程，防止错误和计算机的不当使用（阻止应用相互破坏）
 - ▶ 方便用户使用计算机系统（提供键盘、鼠标、多点触控、图形UI等）
- 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，给用户程序提供服务
 - ▶ 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

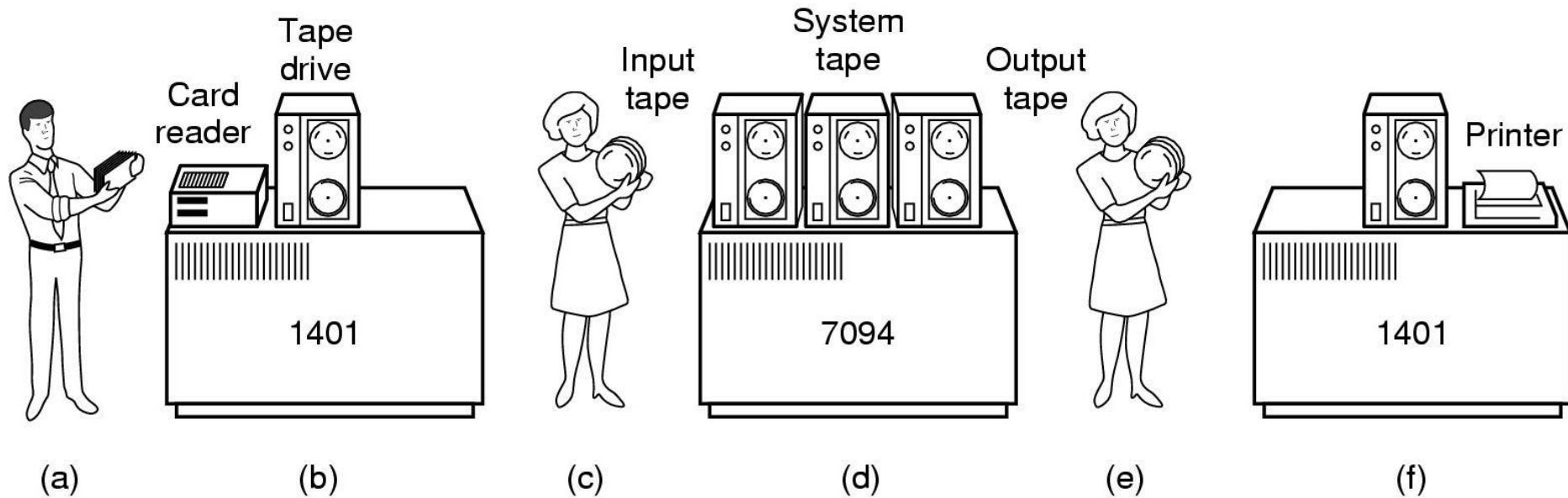
CPU是计算机的心脏
操作系统是计算机的灵魂

操作系统的功能

- Hardware abstraction
 - Provide unified, friendly UI for users
 - Hide the configuration difference of hardware components
- Application integration
 - Provide simple and useful methods for programs designing
 - Schedule and control the running of programs
- Environment management
 - Monitor whole environment of computer
 - Provide a secure, stable and comfortable space for uses



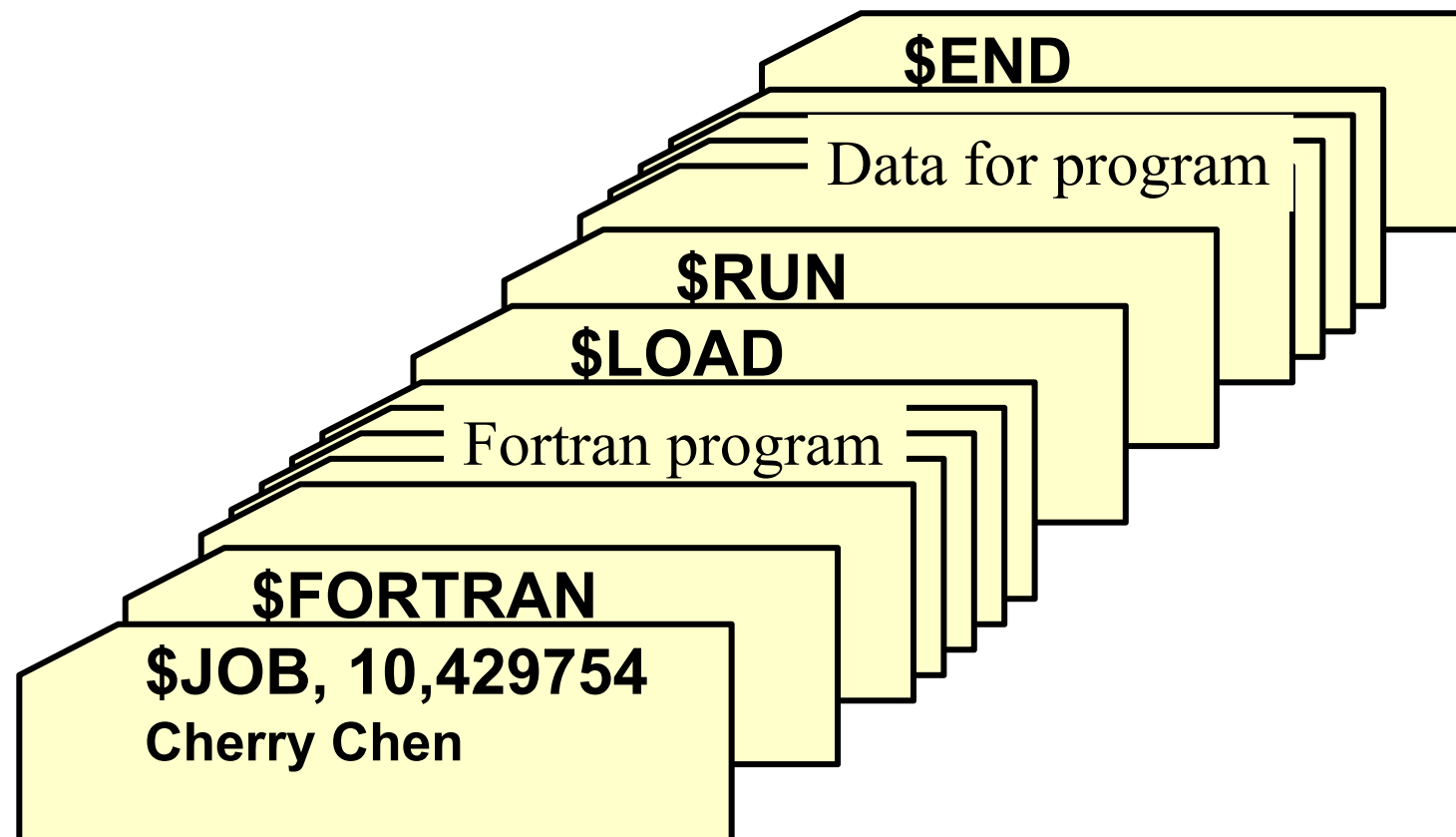
History of Operating Systems (1)



Early batch system

- bring cards to 1401
- read cards to tape
- put tape on 7094 which does computing
- put tape on 1401 which prints output

FMS: Bronze Age



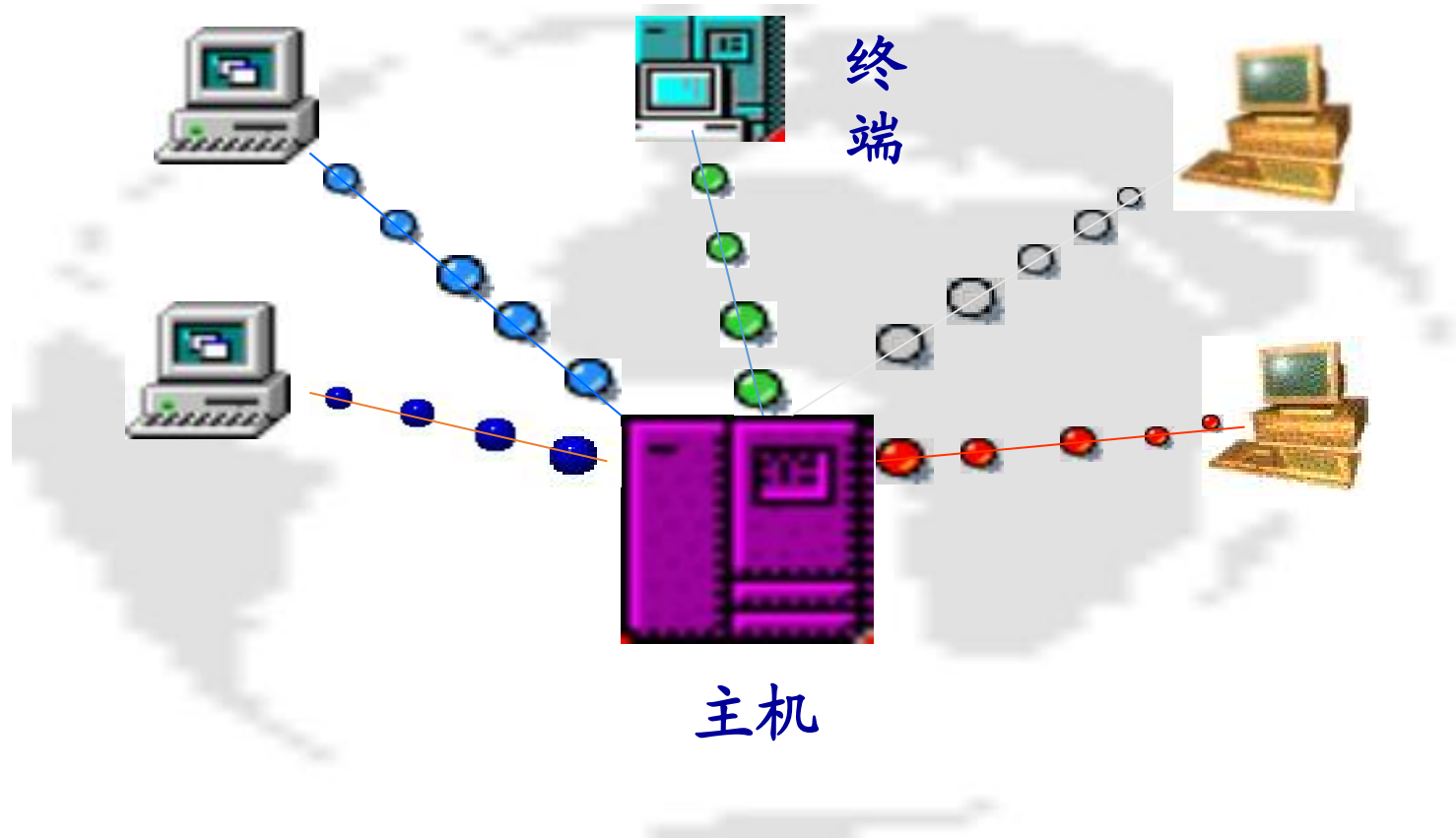
History of OS: Industrial Age

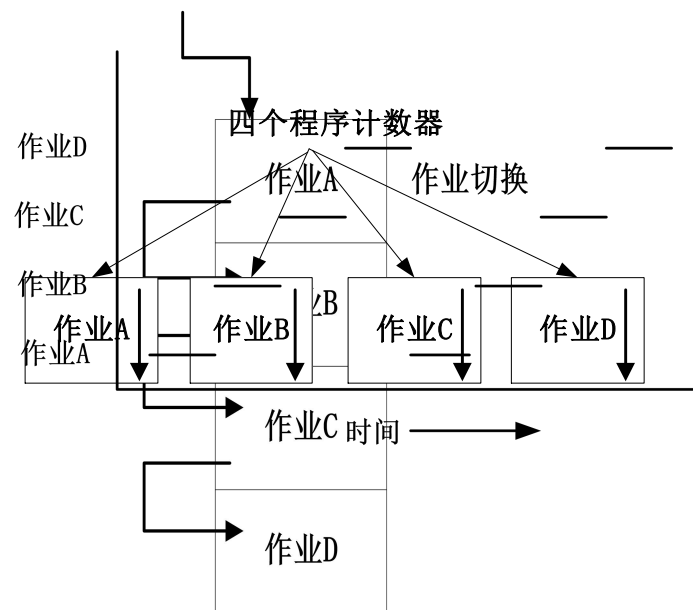
- Multiprogramming System (1965~1980)
 - Famed persons & organizations & OS: Intel, IBM OS/360, Multics, Ken Thompson and Unix, Tanenbaum and Minix, Linus and Linux
 - Industrial standard for computer and OS
- Structure of the operating system
 - Compatible: the beginning of standard
 - Multiprogramming, Spooling, Time-sharing
 - Software engineering
 - Programming language: C, Fortran, Assembler

Multics: seed of modern OS

- Multics (1963~ 1965~ 1969)
 - Multiplexed information and computing services
 - Another software dinosaurs
- Multiprogramming and Time-sharing
 - Allocate CPU time to multi programs in simultaneous way
 - Time-sharing, one mainframe can support 100 online terminal

Multics: seed of modern OS





在操作系统眼中，同一时刻仅处理一个任务

操作系统同时接纳了多个任务，并且在宏观层面实现了任务的并行

分时复用：同一时刻仅处理一个任务，但多个任务交替执行

Ken Thompson: I want to be

- brilliantly

- Multics
- B language
- QED in
- Turing
- Kanai A



1969

or to

ss co

tional Me



- Minix and Linux

- Andrew S.Tanenbaum (1980): minix
- Linux(1991): Linus's minix

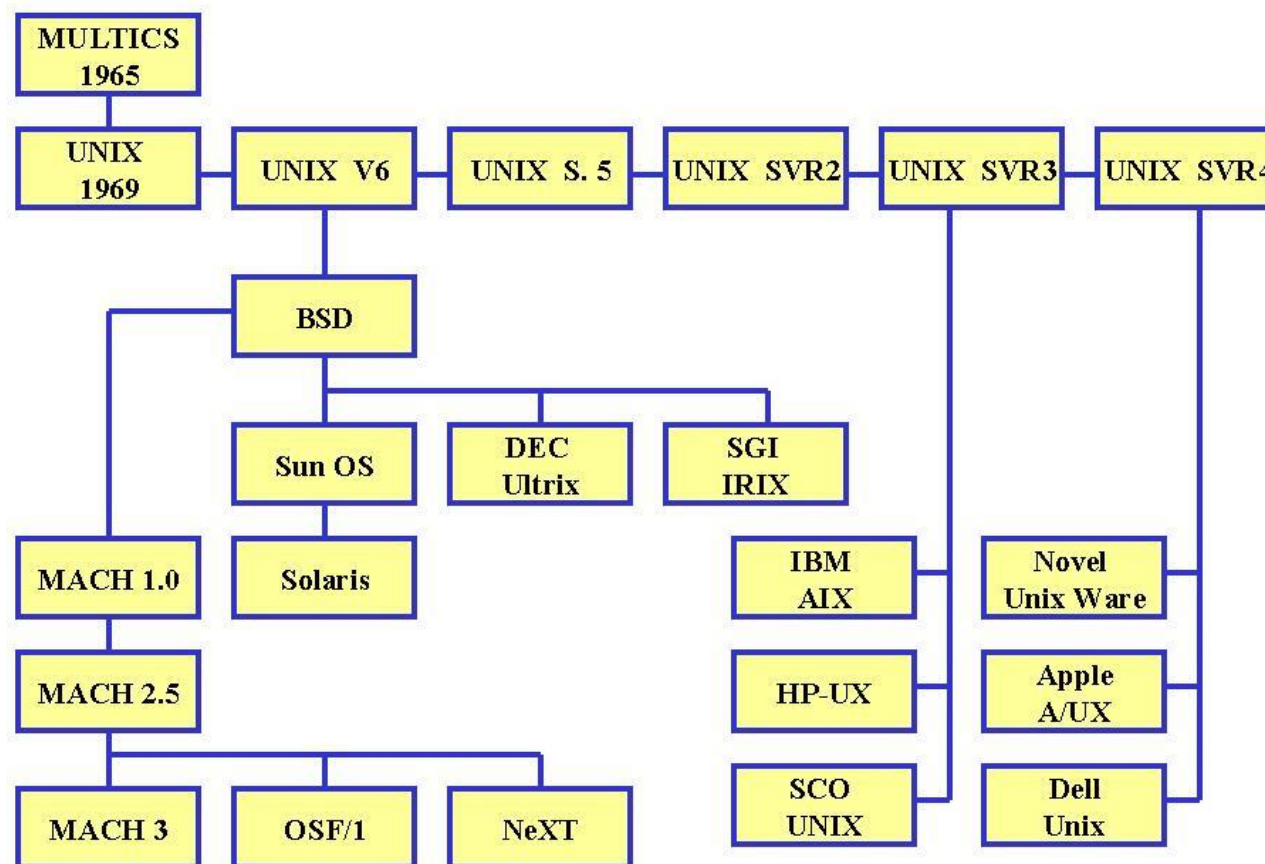


tomu

Ken Thompson



Unix family: more popular than CN&KN



Linus Torvalds: Just do it

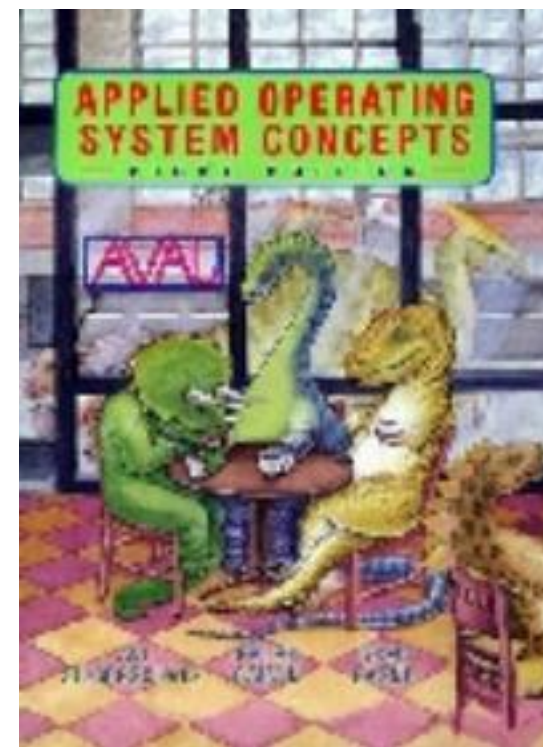
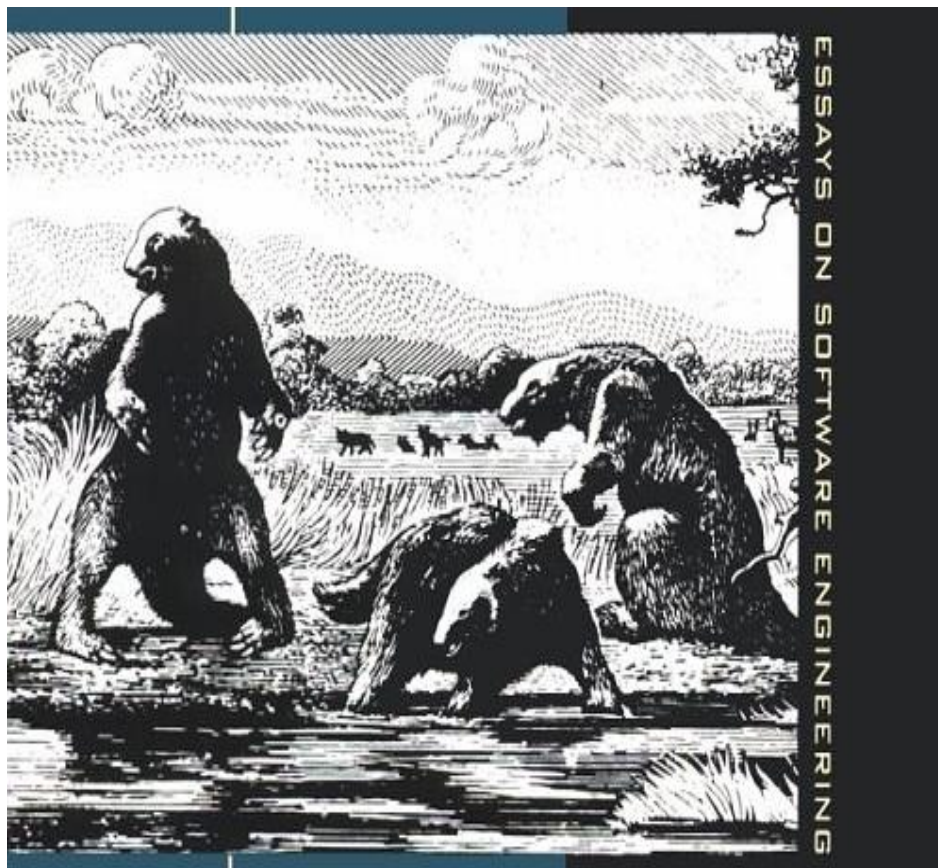
- Linus's motto
 - He created Linux
 - DIY: do it yourself
- Modern computing
 - Open Model
 - Team Work
- The essence of Linux
 - Compatibility
 - Efficiency and security is the most important



Compatible and Soft Engineering

- Compatible
 - Different purpose need different structures, different OS
 - It is too expensive for any company to maintain distinct and totally incompatible produce line
 - One produce line, series product, different functions
- Software engineering
 - Compatible VS Integration
 - No way to write a software to meet conflicting requirements
 - Software dinosaurs: too complex, too enormous
 - IBM OS/360, released with more than 1000 bugs

Compatible and Soft Engineering



History of OS: Golden Age

- Personal Computer & Network (1981~)
 - Famed persons & organizations & OS: Gary Kildall and CP/M, Bill Gates and DOS, Steve Jobs and Mac, IBM OS/2, Open Source and GPL, Embedded system
 - Digital world based on computing
- Structure of the operating system
 - GUI: virtual world constructed by computer
 - Distributed and Sharing: C/S, B/S, Grid...
 - Multiprogramming and parallel
- Computer & Network: new world, new life



现代操作系统所提供的服务

○ 资源管理的抽象化

- 所有硬件资源由操作系统统一管理，用户或者应用程序借助相对标准的**UI**界面或者**API**接口使用资源，不必关心硬件差异的细节

○ 资源的共享与高效利用

- 借助多进程、多线程等模式实现资源的分时复用，通过合理的调度算法设计提升资源利用率

○ 权限隔离

- 通过特权态与用户态的隔离、用户之间地址空间的隔离、用户之间权限的划分实现资源的高效利用

感谢阅读
