

Chapter 1: Introduction

Operating Systems: What and Why?

What is Operating Systems?

2

The concept of operating system is pretty straightforward:

A piece of software that manages a computer, making computer's hardware resources accessible to software through a consistent set of interfaces.

An OS is usually designed for

1. Abstracting away the hardware
抽象
2. Providing a consistent set of interfaces
3. Creating over entire ecosystems
生态系统

What is Operating Systems?

3

Some Concepts with Operating Systems

- Mac、Linux、Windows、Android、iOS
- 虚拟化
 - 内存、cpu、设备、网络
- 分布式对象
- 分布式内存/文件
- 规模计算
 - Mapreduce
 - 数据中心
- 实时和多媒体

Why Study Operating Systems?

4

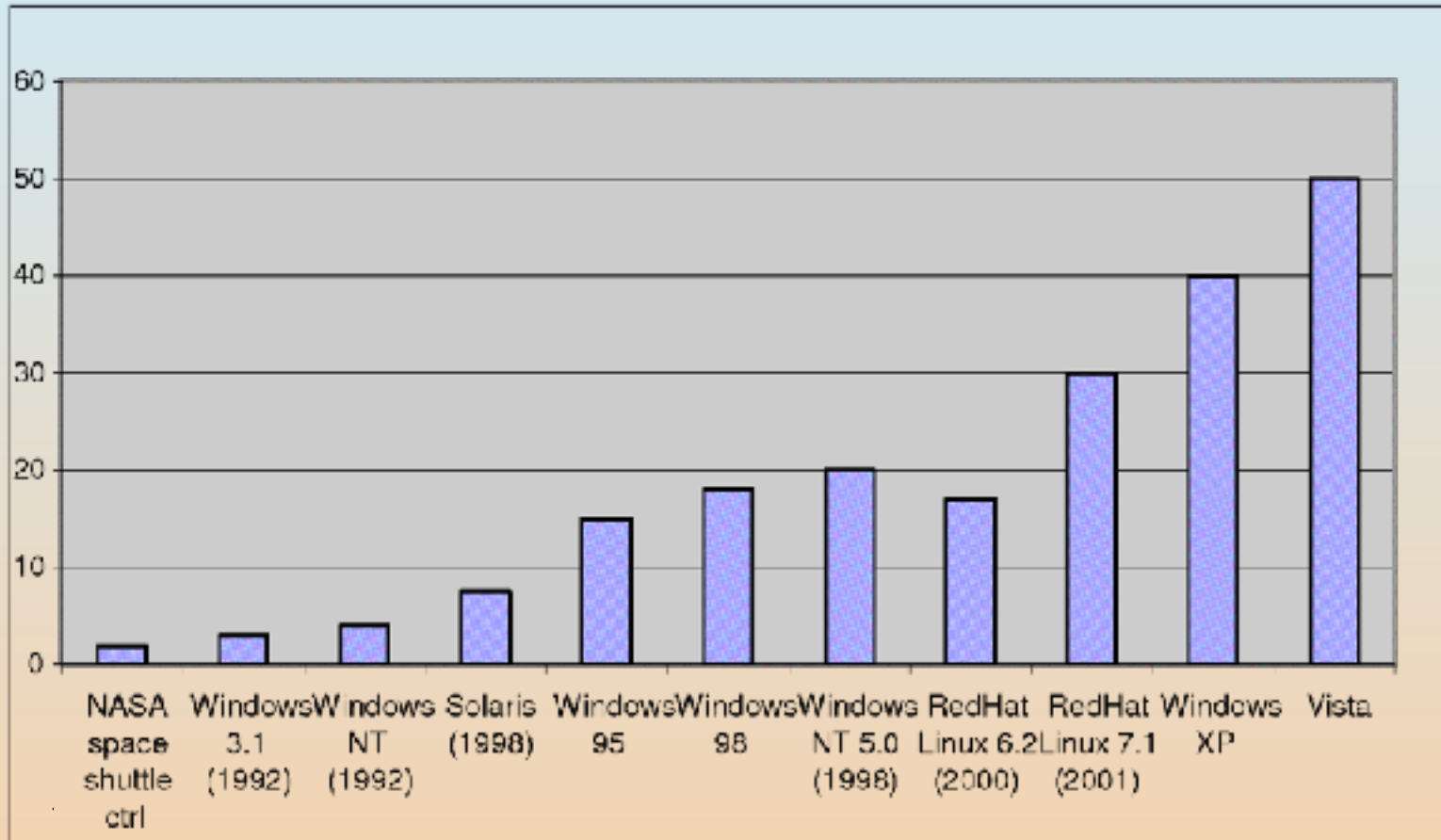
- Run the computer in an efficient manner
- How a program is structured so that an operating system can run.
- A big picture of how each layer of a computer is related to the other, from hardware to software.
-

授人以鱼，不如授人以渔

Increasing Software Complexity

5

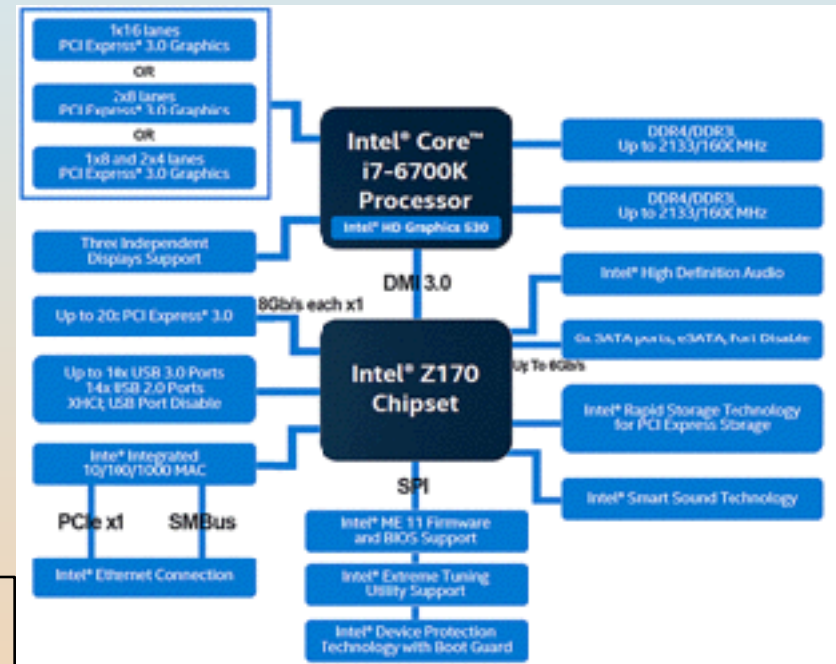
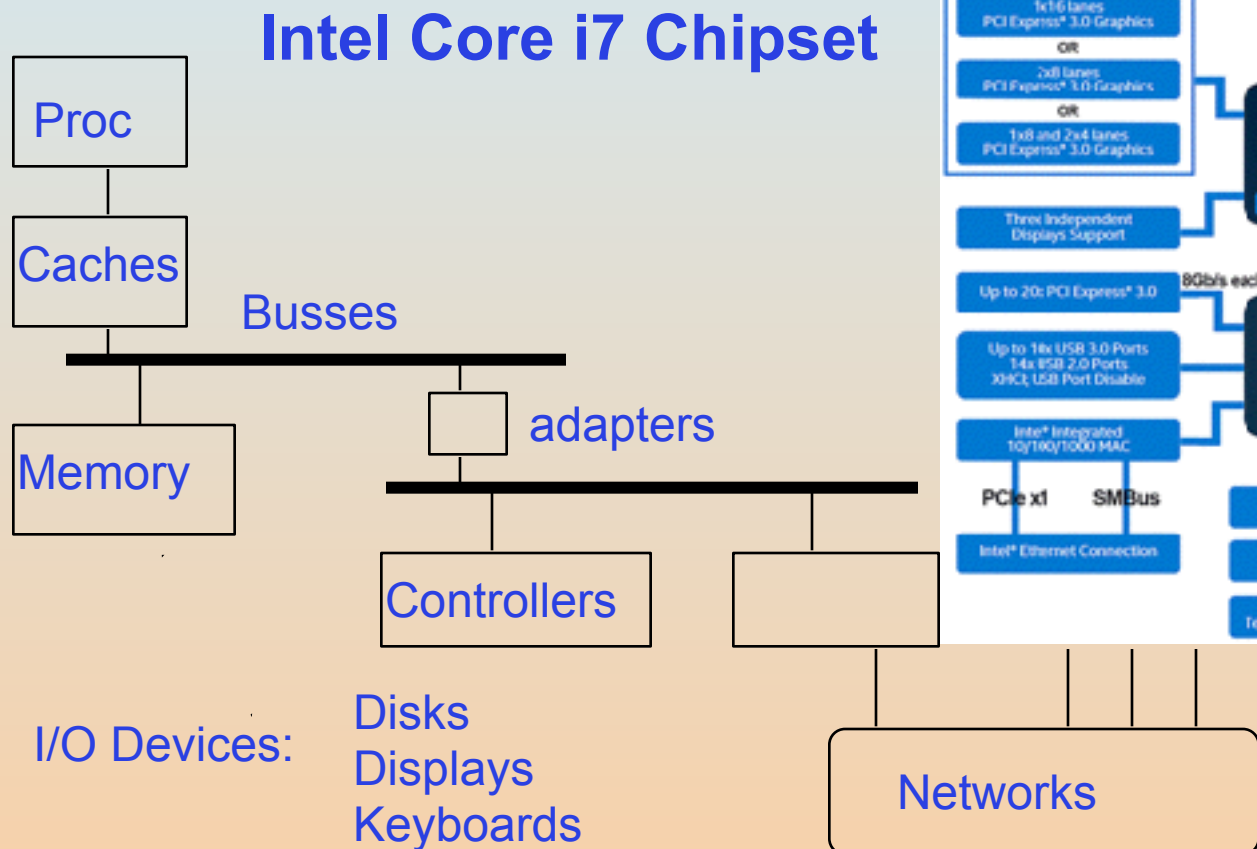
Millions of lines of
source code



From MIT's 6.033 course

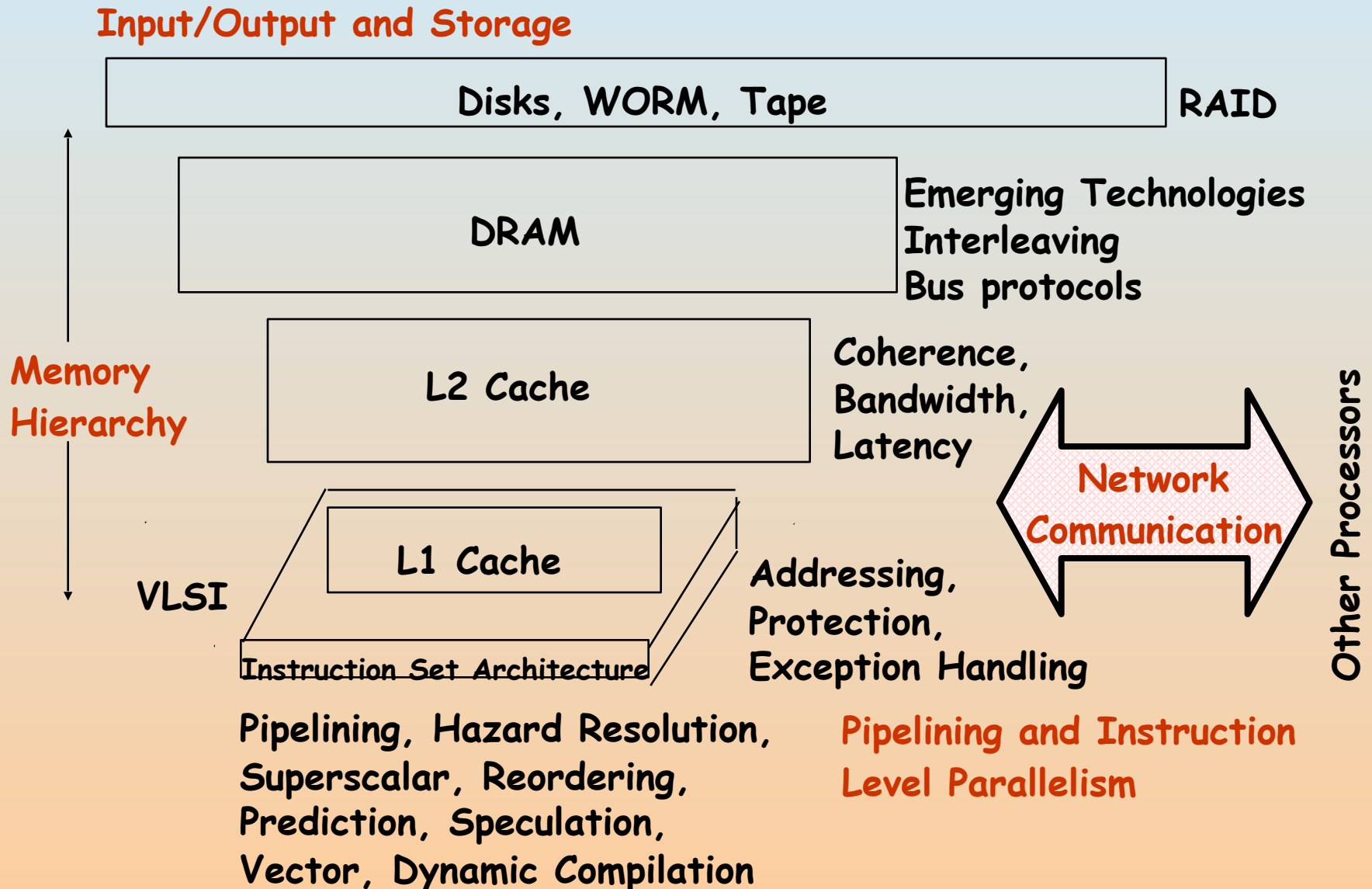
Functionality comes with great complexity!

6



Sample of Computer Architecture Topics

7



Operating Systems

8

- 中央处理机（CPU）
- 内存
- I/O设备（键盘、显示器、鼠标）
- 外部设备（打印机、扫描仪、MODEM 等）
- 外存设备（硬盘、软盘、光盘、磁带等）

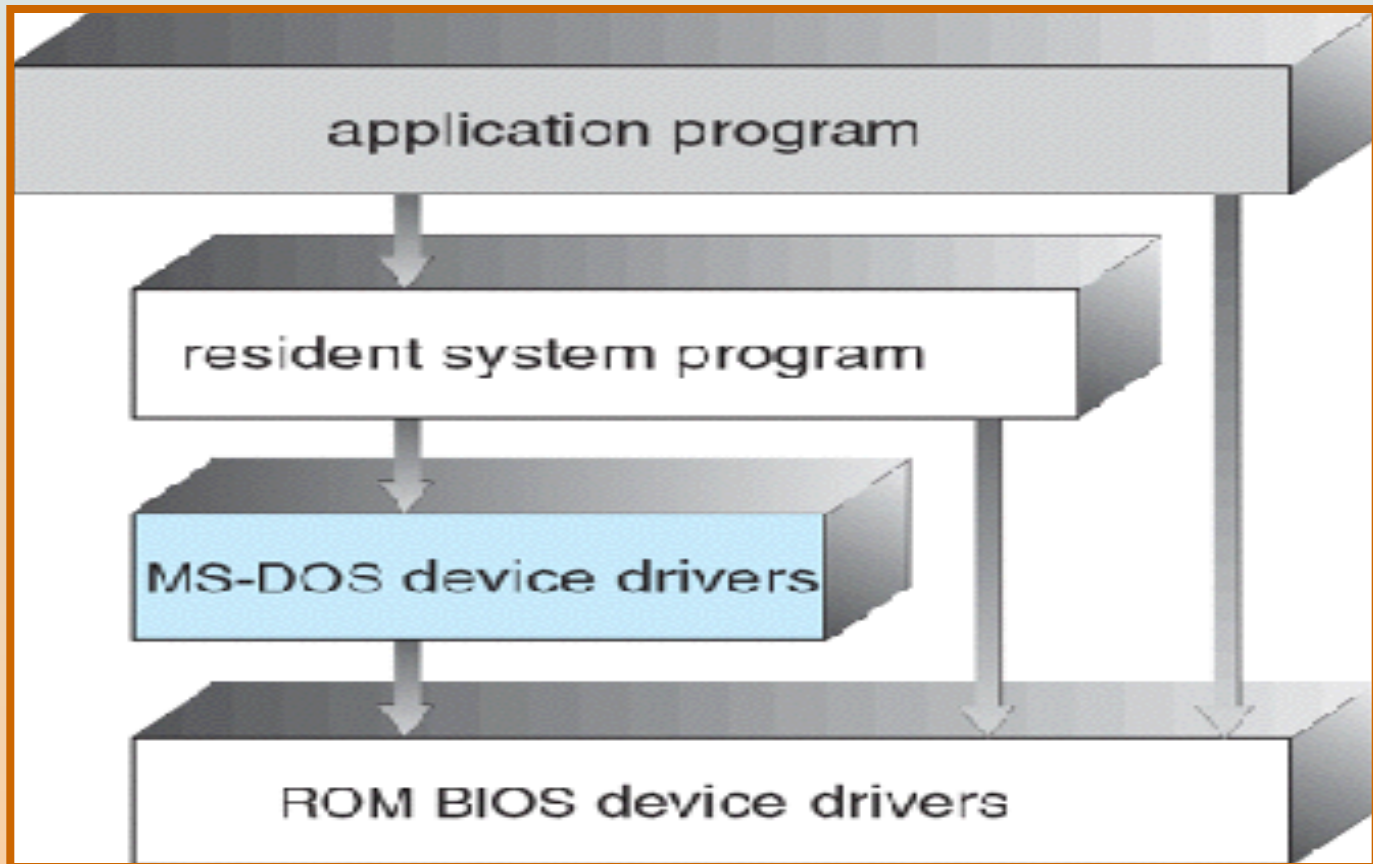
What if we didn't have an Operating System?⁹

- 人工操作
 - 用户独占全机
 - CPU等待人工操作
- 脱机输入/输出

What if only one application?

10

MS-DOS



What if only one application?

11

- 单道批处理系统
 - 自动性
 - 顺序性
 -
 - 单道性
- 资源（cpu,内存,I/O设备）利用率？
- 系统吞吐量？

More complex OS: Multiple Apps

12

- Multi-programming (多道程序)
- Problem: Run multiple applications in such a way that they are protected from one another
- Goal:
 - Keep User Programs from Crashing OS
 - Keep User Programs from Crashing each other
 - [Keep Parts of OS from crashing other parts?]
- (Some of the required) Mechanisms:
 - Address Translation
 - Dual Mode Operation
- Simple Policy:
 - Programs are not allowed to read/write memory of other Programs or of Operating System

Multi-programming (多道程序)

13

- 多道性
- 无序性
- 调度性
- 平均周转时间?
- 交互能力?
- 处理机/内存/I/O设备/文件/...管理?

Time-Sharing Systems–Interactive Computing

多路

- The CPU is multiplexed among several jobs that are kept in memory and on disk (the CPU is allocated to a job only if the job is in memory)
- A job swapped in and out of memory to the disk
- On-line communication between the user and the system is provided
 - When the operating system finishes the execution of one command, it seeks the next “control statement” from the user’s keyboard
- 多路性
- 独立性
- 及时性
- 交互性

Real-Time Systems

15

专用

- Often used as a control device in a dedicated application such as controlling scientific experiments, medical imaging systems, industrial control systems, and some display systems
- Well-defined fixed-time constraints 限制
- Real-Time systems may be either hard or soft real-time
- 实时控制
- 实时信息处理
- 可靠性

Real-Time Systems (Cont.)

16

- Hard real-time:
 - Secondary storage limited or absent, data stored in short term memory, or read-only memory (ROM)
 - Conflicts with time-sharing systems, not supported by general-purpose operating systems
- Soft real-time
 - Limited utility in industrial control of robotics
 - Integrate-able with time-share systems
 - Useful in applications (multimedia, virtual reality) requiring tight response times

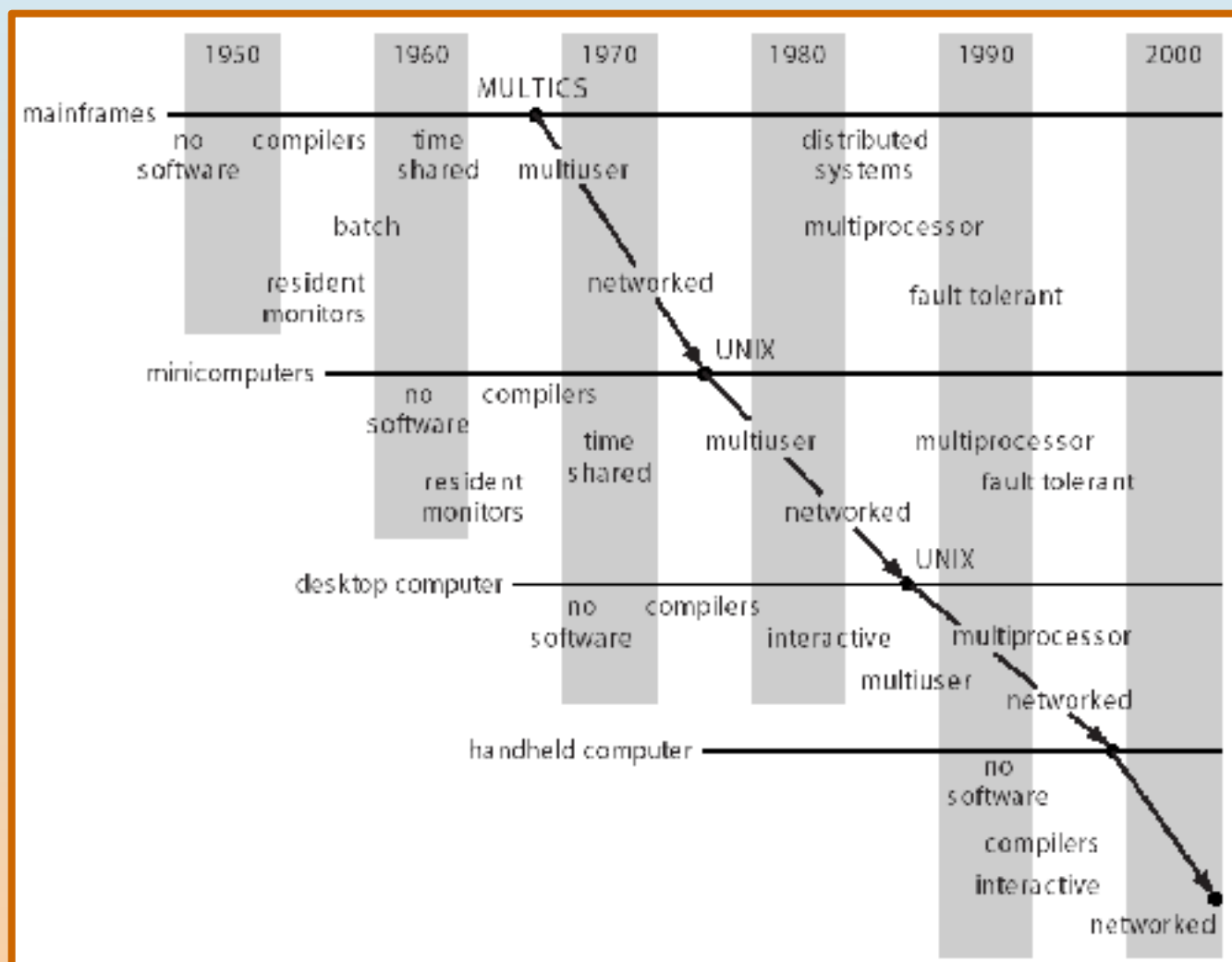
Handheld Systems

17

- Personal Digital Assistants (PDAs)
- Cellular telephones
- Issues:
 - Limited memory
 - Slow processors
 - Small display screens

Migration of Operating-System Concepts and Features

18



OS Characteristics

19

- 并发(Concurrence)
- 共享(Sharing) ☐
- 虚拟(Virtual)
- 异步(Asynchronism)

Why Study Operating Systems?

20

- Learn how to build complex systems:
 - How can you manage complexity for future projects?
- Engineering issues:
 - Why is the web so slow sometimes? Can you fix it?
 - What features should be in the next mars Rover?
 - How do large distributed systems work? (Kazaa, etc)

Why Study Operating Systems?

21

- Buying and using a personal computer:
 - Why different PCs with same CPU behave differently
 - How to choose a processor (Opteron, Itanium, Celeron, Pentium, Hexium)? [Ok, made last one up]
 - Should you get Windows XP, 2000, Linux, Mac OS ...?
 - Why does Microsoft have such a bad name?
- Business issues:
 - Should your division buy thin-clients vs PC?
- Security, viruses, and worms
 - What exposure do you have to worry about?

Goals for Today

22

- 操作系统的发展
- 操作系统的功能
- 操作系统的特征
- 操作系统的性能指标
- 现代操作系统设计及其基本问题

操作系统的发展

23

- 批处理操作系统
- 分时操作系统
- 实时操作系统
- 网络操作系统
- 并行操作系统
- 分布式操作系统
- 嵌入式操作系统
- 一些典型操作系统

操作系统的发展

24

- 批处理操作系统
 - 主要特征：
 - 用户脱机工作
 - 成批处理作业
 - 作业周转时间长
 - 单道：自动，顺序，单道
 - 多道：调度，无序，多道
- 分时操作系统
- 实时操作系统

操作系统的发展

25

- 批处理操作系统
- 分时操作系统
 - 分时技术 (Time-sharing)
 - 多道程序技术 (Multi-programming)
 - 主要特征：
 - 多道
 - 独立
 - 交互
 - 及时
- 实时操作系统

操作系统的发展

26

- 批处理操作系统
- 分时操作系统
- 实时操作系统
 - 多道
 - 独立
 - 交互
 - 及时
 - 可靠

操作系统的发展

27

- 网络操作系统
 - Centralized
 - Client/Server
 - Peer2Peer(P2P)
 - 主要功能：
 - 具备网络通信能力
 - 提供各种网络服务
 - 通常操作系统应具备的功能
 - 主要特征：
 - 资源共享
 - 独立自主

并行操作系统

28

- 运行在并行计算机上的操作系统
- 并行处理技术
 - 提高同一时间间隔内的操作数量
 - 时间并行
 - 空间并行
 - 数据并行
 - 任务并行
- 并行计算
 - Google 搜索引擎
 - 曙光 3000

分布式操作系统

29

- 通过网络连接在一起的若干计算机的集合，有各自的局部存储器和外部设备。从硬件上讲，它与计算机局域网没有任何区别，主要区别在于软件。
- 主要特征：
 - 独立，无主从关系
 - 协作
 - 数据/任务分布未知
 - 健壮性
- 主要功能
 - 多机进程通信；
 - 分布资源共享；
 - 并行分布计算；
 - 分布式网络管理。

嵌入式操作系统

30

- 嵌入式系统：嵌入在各种设备、装置或系统（非“计算机”）中，完成特定功能的软硬件系统
- 运行在嵌入式系统环境中的操作系统
- 主要特征
 - 微型化
 - 可定制
 - 实时性
 - 可靠性
 - 易移植性
 - 开发环境

典型操作系统

31

- OS/360操作系统：通用操作系统，兼容性
- MULTICS操作系统：首次应用许多现代操作系统领域概念雏形
- UNIX操作系统
- MS/DOS
- MS/Windows
- MAC OS

操作系统的功能

32

- CPU管理
- 存储管理
- 文件管理
- 设备管理
- 用户接口
- 网络与通信管理

操作系统的功能

33

- CPU管理
 - 进程/线程控制和管理
 - 进程同步和互斥 (mutual exclusion)
 - 进程通信和死锁 (dead lock)
 - 处理器调度，作业调度和进程调度

操作系统的功能

34

- 存储管理
 - 存储分配
 - 存储共享
 - 存储保护
 - 地址转换
 - 存储扩充

操作系统的功能

35

- 文件管理
 - 目录管理
 - 存取控制/保护
 - 逻辑组织
 - 物理组织
 - 文件存储空间管理

操作系统的功能

36

- 设备管理
 - 设备分配
 - 设备驱动
 - 缓冲管理

操作系统的功能

37

- 用户接口
 - 命令接口
 - 程序接口
 - 图形接口

操作系统的特征

38

- 并发(Concurrence)
- 共享(Sharing) **?**
- 虚拟(Virtual)
- 异步(Asynchronism)

操作系统的特征

39

- 并发(Concurrence)
 - 如何从一个活动切换到另一个活动?
 - 怎样将各个活动隔离开来，使之互不干扰，免遭对方破坏?
 - 怎样让多个活动协作完成任务?
 - 怎样协调多个活动对资源的竞争？如何保证每个活动的资源不被其它进程侵犯？
 - 多个活动共享文件数据时，如何保证数据的一致性？

操作系统的特征

40

- 共享(Sharing)

- 互斥共享:

- 同时访问

操作系统的性能指标

41

- 系统的可靠性
- 系统吞吐率
- 系统的响应时间
- 系统资源的利用率
- 可移植性

现代操作系统设计及其基本问题

42

- Conflict: 解决冲突的策略设计;
- Coordination: 协调协作活动的关系;
- Coherence: 保证数据的一致性;
- Access Control: 实现数据存取控制。

现代操作系统设计及其基本问题

43

- 用户行为难以预料
 - No user action should crash the machine
 - Can't just flee when the file system fills up
- 种种破坏性可能
 - 0作为分母, 越权访问
 - Users will try to use 130% of everything
 - 非法拷贝别人的文件
 - 删除文件或文件系统

现代操作系统设计及其基本问题

44

- 硬件问题难以预见
 - 硬盘出现坏道（扇区）
 - 硬盘损坏
 - 内存被宇宙射线修改
- 系统的健壮性、容错性
 - OS should run “forever”
 - 各类错误随时间的累积