第 1 讲：Advanced OS Overview

第三节：Rethink OS Components

陈渝

清华大学计算机系

yuchen@tsinghua.edu.cn

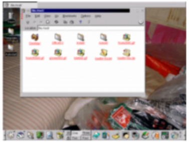
2020 年 2 月 15 日

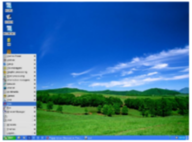
History of OS: Change!

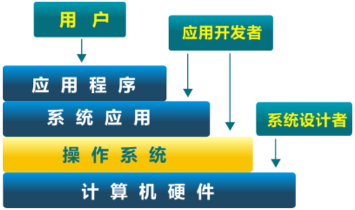
Porperty	Element	1980	2020	Factor
Speed	CPU	1 MIPS	97,000 MIPS	9.7×10^5
	Memory	500 ns	0.4 ns	3.6×10^3
	Disk	18 ms	4 ms	2.2×10^1
	Network	300 b/sec	100 Gb/sec	3.3×10^8
Capacity	Memory	64 KB	1 TB	2.5×10^5
	Disk	1 MB	4 TB	4.0×10^6
Other	Address bits	8	64	8
	Users/machine	10	0.1	1.0×10^{-2}
Cost	Per MIP	$100K/MIP	$.007899	1.0×10^7

Roles of the OS

- OS depends on
  - available hardware and software
  - changing uses of machines
  - changing expectations of users







用户

应用开发者

应用程序

系统应用

系统设计者

操作系统

计算机硬件

OS Concepts

- Single-Machine OS/VMM
    - Memory management
    - Process management
    - Synchronization
    - File systems and device support

OS Concepts

- Single-Machine OS/VMM
  - Memory management
  - Process management
  - Synchronization
  - File systems and device support
- Security
- Correctness
- Distributed System

Single-Machine OS/VMM

- Purposes
  - Clean virtual machine
  - Hardware independence
  - Resource sharing and management
  - Persistent Data Storage
  - Protection
  - Real time support
  - Parallelism
  - human interface

Single-Machine OS/VMM

- Purposes
  - Clean virtual machine
  - Hardware independence
  - Resource sharing and management
  - Persistent Data Storage
  - Protection
  - Real time support
  - Parallelism
  - human interface

- Strategy
  - How do we organize the OS effectively for development, evolution, performance, and security?
  - How do we use multi- processor machines effectively?

Single-Machine: Memory Management

- Purposes
  - Virtual memory: provides the illusion of infinite physical memory
  - Swapping: moves processes to disk as necessary
  - Paging: allows processes to run with only the active pages in memory
  - Buffer Cache: speedup the IO access

Single-Machine: Memory Management

- Purposes
  - Virtual memory: provides the illusion of infinite physical memory
  - Swapping: moves processes to disk as necessary
  - Paging: allows processes to run with only the active pages in memory
  - Buffer Cache: speedup the IO access

- Strategy
  - How do we coordinate machines to share memory?
  - How can we simplify memory management as memory becomes abundant?

Single-Machine: Process Scheduling

- Purposes
  - Provides the illusion of multiple processes running at the same time on a single processor
  - Context switching: changing the attention of the processor
    - Involves saving and restoring states
    - Necessary to cross kernel boundary

Single-Machine: Process Scheduling

- Purposes
  - Provides the illusion of multiple processes running at the same time on a single processor
  - Context switching: changing the attention of the processor
    - Involves saving and restoring states
    - Necessary to cross kernel boundary

- Strategy
  - How do we achieve fairness, high throughput, and responsiveness at the same time?
  - How do we reduce or avoid the cost of context switching?

Single-Machine: Synchronization

- Purposes
  - Provides correct execution or coordinating threads in the face of arbitrary context switching

Single-Machine: Synchronization

- Purposes
  - Provides correct execution or coordinating threads in the face of arbitrary context switching

- Strategy
  - Atomic actions: all or nothing
  - Mutual exclusion: one thread in the critical section at a time
  - Semaphores: atomic, counter-based locks
  - Avoid Deadlock: circular waiting on resources

Single-Machine: File Systems

- Purposes
  - File: data + attributes
  - File system services:
    - Organization
    - Naming
    - Access
    - Synchronization
    - Protection and security

Single-Machine: File Systems

- Purposes
  - File: data + attributes
  - File system services:
    - Organization
    - Naming
    - Access
    - Synchronization
    - Protection and security

- Strategy
  - How do we make different file systems work together, even across machines?
  - How do we provide consistency, availability, and reliability to copies of a file across multiple machines?
  - How do we handle very large data sets?

Single-Machine: I/O Device

- Purposes
  - I/O devices tend to be a lot slower than memory speed
  - Caching: stores extra data in memory in hope of near-term reuse

Single-Machine: I/O Device

- Purposes
  - I/O devices tend to be a lot slower than memory speed
  - Caching: stores extra data in memory in hope of near-term reuse

- Strategy
  - How do we coordinate the memory resources across machines to enhance performance?
  - How do we handle new devices with new characteristics?