Understanding Operating Systems
Seventh Edition

Chapter 1

Introducing Operating Systems

Learning Objectives

After completing this chapter, you should be able to describe:

- Innovations in operating systems development
- The basic role of an operating system
- The major operating system software subsystem managers and their functions
- The types of machine hardware on which operating systems run

Learning Objectives (cont'd.)

- The differences among batch, interactive, real-time, hybrid, and embedded operating systems
- Design considerations of operating systems designers

Introduction

- Operating systems
  - Manage computer system hardware and software
- This text explores:
  - What they are
  - How they work
  - What they do
  - Why they do it
- This chapter briefly describes:
  - Workings of operating systems

What Is an Operating System?

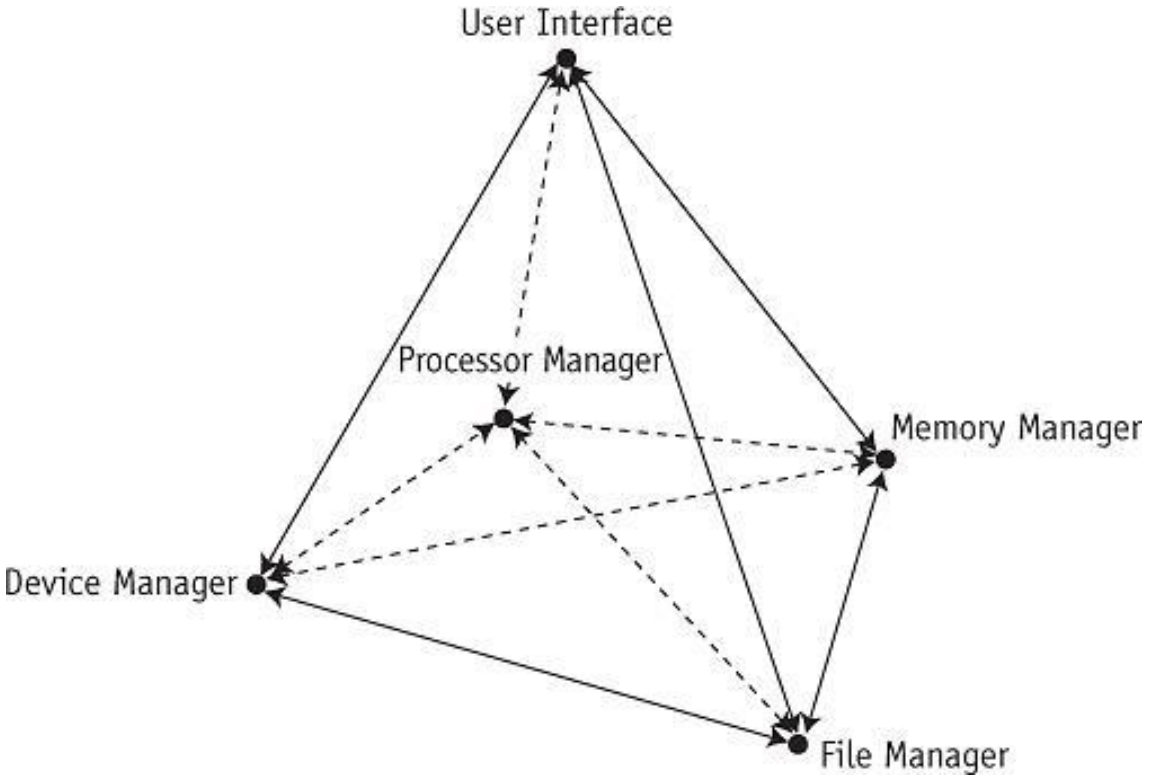
- Computer system
  - Software (programs)
  - Hardware (tangible machine/electronic components)
- Operating system
  - Chief software component
  - Manages all hardware and all software and controls:
    - Every file, device, section of main memory, and moment of processing time
    - *Who* can use the system and h*ow* system is used

Operating System Software

- Includes four essential subsystem managers
  - Memory Manager
  - Processor Manager
  - Device Manager
  - File Manager
- Each manager:
  - Works closely with other managers
  - Performs a unique role

Operating System Software (cont'd.)

(figure 1.1) This pyramid represents an operating system on a stand-alone computer unconnected to a network. It shows the four subsystem managers and the user interface.

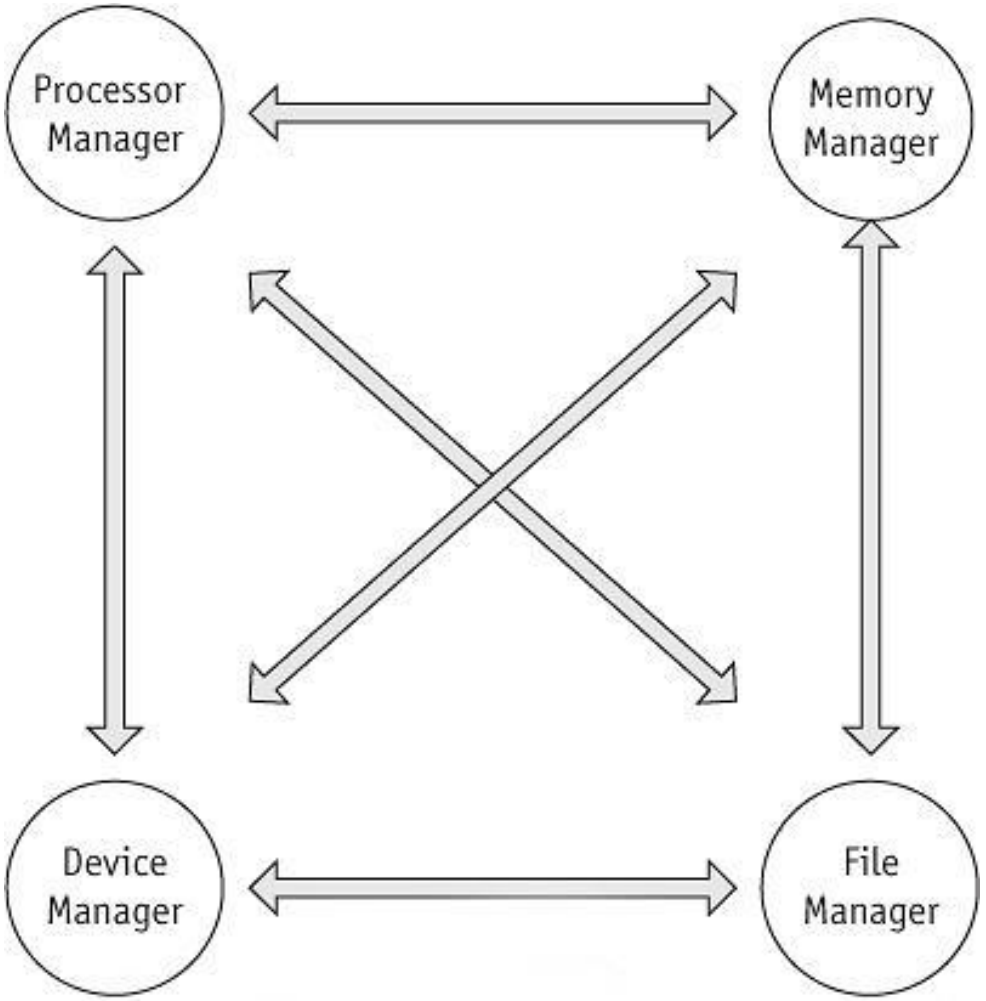


Operating System Software (cont'd.)

- User Interface
  - Allows the user to issue commands to the operating system
- Manager tasks
  - Monitor the system’s resources continuously
  - Enforce policies determining:
    - Who gets what, when, and how much
  - Allocate a resource (when appropriate)
  - Deallocate a resource (when appropriate)

(figure 1.2)

Each manager at the base of the pyramid takes responsibility for its own tasks while working harmoniously with every other manager.



Operating System Software (cont'd.)

- Network Manager
  - Coordinates the services required for multiple systems to work cohesively together
    - Shared network resources: memory space, processors, printers, databases, applications, etc.

Main Memory Management

- In charge of main memory
  - Random access memory (RAM)
    - Requires constant flow of electricity to hold data
- Responsibilities include:
  - Checking validity and legality of memory space request
  - Reallocating memory to make more useable space available
  - Deallocating memory to reclaim it
  - Protecting space in main memory occupied by operating system

Main Memory Management

- Read-only memory (ROM)
  - Another type of memory
  - Critical when computer is powered on
  - Holds firmware: programming code
    - When and how to load each piece of the operating system after the power is turned on
  - Non-volatile
    - Contents retained when the power is turned off

Processor Management

- In charge of allocating Central Processing Unit (CPU)
- Tracks process status
  - Program's "instance of execution"
- Comparable to a traffic controller
  - When a process is finished or maximum computation time expired
    - Processor Manager reclaims the CPU and allocates to next waiting process
  - Computer with multiple CPUs
    - More complex management

Device Management

- In charge of connecting with every available device
  - Printers, ports, disk drives, etc.
- Responsibilities include:
  - Choosing most efficient resource allocation method
    - Based on scheduling policy
  - Identifying each device uniquely
  - Starting device operation (when appropriate)
  - Monitoring device progress
  - Deallocating the device
- What is the function of a device driver?

File Management

- In charge of tracking every file in the system
  - Data files, program files, compilers, application programs, etc.
- Responsibilities include:
  - Enforcing user/program resource access restrictions
    - Uses predetermined access policies
  - Controlling user/program modification restrictions
    - Read-only, read-write, create, delete
  - Allocating space for a file on secondary storage
    - One large storage area or smaller linked pieces
  - Retrieving files efficiently

Network Management

- Included in operating systems with networking capability
- Authorizes users to share resources
  - Overall responsibility for every aspect of network connectivity
    - Devices, files, memory space, CPU capacity, etc.

User Interface

- Portion of the operating system
  - Direct interaction with users
- Two primary types
  - Graphical user interface (GUI)
    - Input from pointing device
    - Menu options, desktops, and formats vary
  - Command line interface
    - Keyboard-typed commands that display on a monitor
    - Strict requirements for every command: typed accurately; correct syntax; combinations of commands assembled correctly

Cooperation Issues

- No single manager performs tasks in isolation
- Each element of an operating system
  - Performs individual tasks *and*
  - Harmoniously interacts with other managers
    - Incredible precision required for operating system to work smoothly
    - More complicated when networking is involved

Cloud Computing

- Practice of using Internet-connected resources
  - Performing processing, storage, or other operations
- Operating system maintains responsibility
  - Managing all local resources and coordinating data transfer to and from the cloud
- Role of the operating system
  - Accessing resources
  - Managing the system efficiently

An Evolution of Computing Hardware

- Hardware: physical machine and electronic components
  - Main memory (RAM)
    - Data/Instruction storage and execution
  - Central processing unit (CPU)
    - Controls interpretation and execution of instructions
    - Initiates or performs these operations: storage, data manipulation and input/output
  - Input/output devices (I/O devices)
    - All peripheral devices in the system: printers, disk drives, CD/DVD drives, keyboards, etc.

An Evolution of Computing Hardware (cont'd.)

- Computer classification
  - At one time: based on memory capacity
- Current platforms

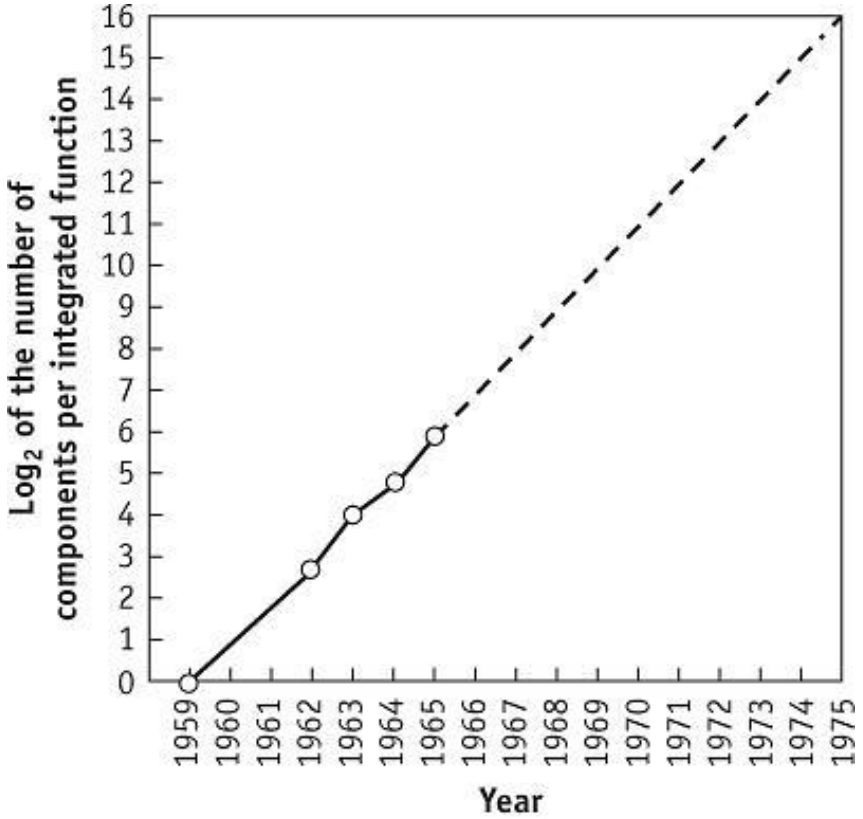
Platform	Operating System
Telephones, tablets	Android, iOS, Windows
Laptops, desktops	Linux, Mac OS X, UNIX, Windows
Workstations, servers	Linux, Mac OS X Server, UNIX, Windows Server
Mainframe computers	Linux, UNIX, Windows, IBM z/OS
Supercomputers	Linux, UNIX

(table 1.1)
A brief list of platforms and a few of the operating systems designed to run on them, listed in alphabetical order.

An Evolution of Computing Hardware (cont'd.)

- Moore’s Law: Gordon Moore, 1965
  - Each new processor chip compared to its predecessor
    - Twice as much capacity
    - Released within 18-24 months

An Evolution of Computing Hardware (cont'd.)



Number of components per integrated function for minimum cost per component extrapolated vs time.

(figure 1.6)

Gordon Moore's 1965 paper included the prediction that the number of transistors incorporated in a chip will approximately double every 24 months [Moore, 1965].

Courtesy of Intel Corporation.

Types of Operating Systems

- Five categories
  - Batch
  - Interactive
  - Real-time
  - Hybrid
  - Embedded
- Two distinguishing features
  - Response time
  - Method of data entry into the system

Types of Operating Systems (cont'd.)

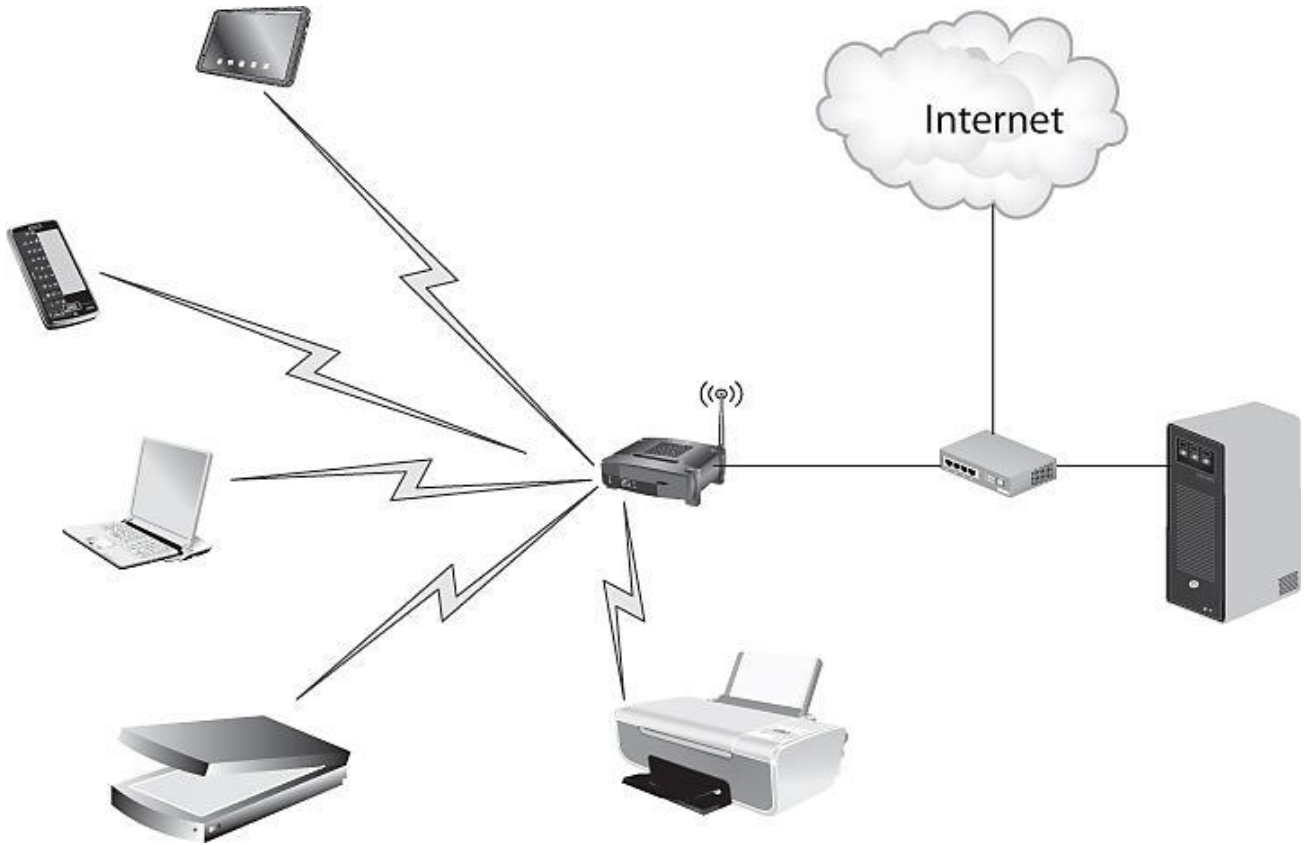
- Batch systems: jobs entered as a whole and in sequence
  - Input relied on punched cards or tape
  - Efficiency measured in throughput
- Interactive systems: allow multiple jobs
  - Faster turnaround than batch systems
  - Slower than real-time systems
  - Introduced to provide fast turnaround when debugging programs
  - Complex algorithms: share processing power

Types of Operating Systems (cont'd.)

- Real-time systems
  - Reliability is critical
  - Used in time-critical environments
    - Spacecraft, airport traffic control, fly-by-wire aircraft, critical industrial processes, medical systems, etc.
  - Two types of real-time systems
    - Hard real-time systems: risk total system failure if the predicted time deadline is missed
    - Soft real-time systems: suffer performance degradation as a consequence of a missed deadline

Types of Operating Systems (cont'd.)

- Hybrid systems
  - Combination of batch and interactive
  - Light interactive load
    - Accepts and runs batch programs in the background
- Network operating systems
  - Special class of software
    - Users perform tasks using few, if any, local resources, e.g., cloud computing
  - Wireless networking capability
    - Standard feature in many computing devices: cell phones, tablets, and other handheld Web browsers



(figure 1.7)

Example of a simple network. The server is connected by cable to the router and other devices connect wirelessly.

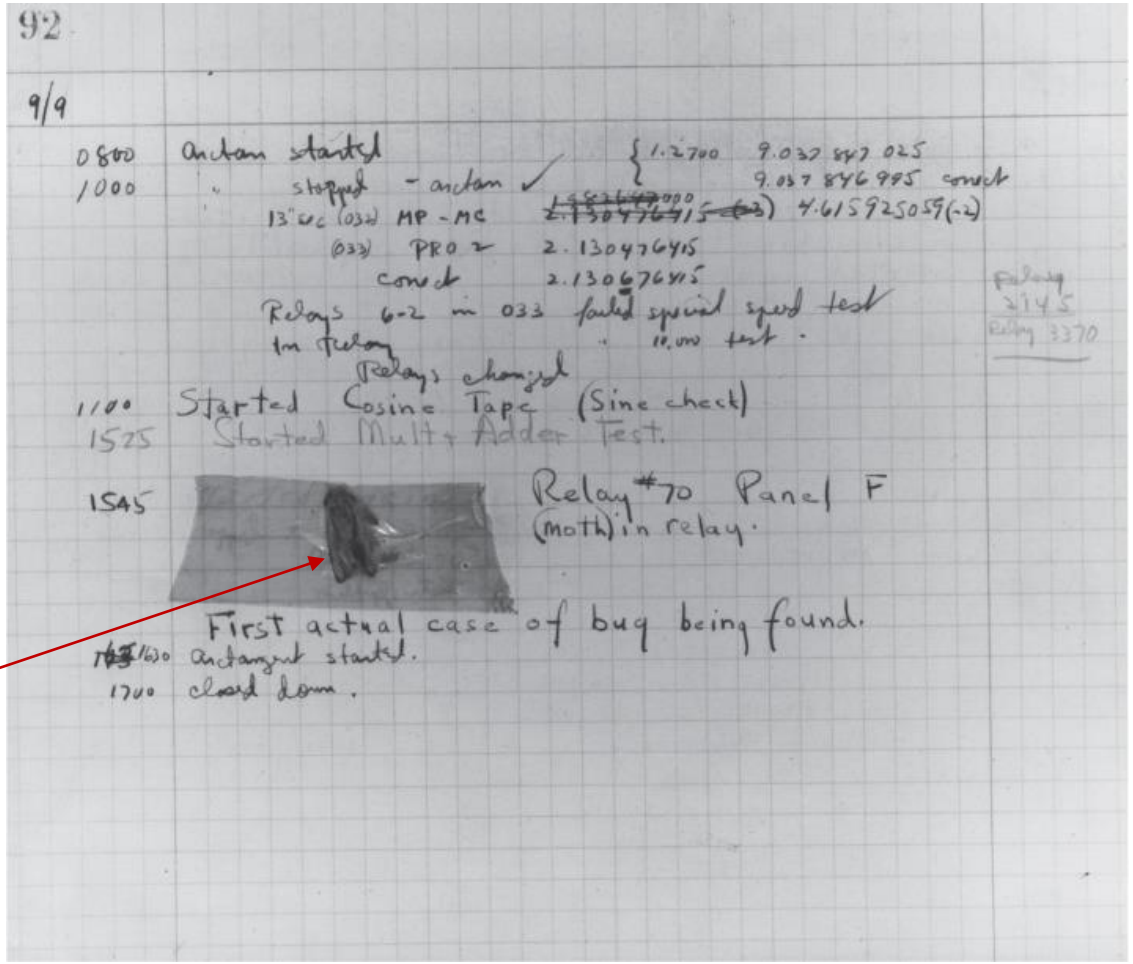
Types of Operating Systems (cont'd.)

- Embedded systems
  - Computers placed inside other products
    - Automobiles, digital music players, elevators, pacemakers, etc.
  - Adds features and capabilities
  - Operating system requirements
    - Perform specific set of programs
    - Non-interchangeable among systems
    - Small kernel and flexible function capabilities

Brief History of Operating Systems Development

- Evolution of operating systems: parallels evolution of computer hardware
- 1940s
  - Computers operated by programmers presiding from the main console
  - Fixing a program error
    - 1) stop the processor; 2) read register contents; 3) make corrections in memory; 4) resume operations
  - Poor machine utilization
    - CPU processed data and performed calculations for fraction of available time
  - “Bug”: system not working correctly

(figure 1.8)
Dr. Grace Hopper's research journal included the first computer bug, the remains of a moth that became trapped in the computer's relays, causing the system to crash. [Photo © 2002 IEEE]



Brief History of Operating Systems Development (cont'd.)

- 1950s: second generation
  - Two widely adopted improvements
    - Computer operators: humans hired to facilitate machine operation
    - Job scheduling: programs with similar requirements grouped together (batch)
  - Expensive time lags between CPU and I/O devices
  - Improvements to the CPU and the system
    - I/O devices (tape drives and disks): gradually became faster

Brief History of Operating Systems Development (cont'd.)

- 1950s
  - Improvements to the CPU and the system
    - Records blocked before retrieval or storage
    - Control unit placed between I/O and CPU as a buffer: reduced speed discrepancy
- 1960s
  - Faster CPUs
  - Speed caused problems with slower I/O devices

Brief History of Operating Systems Development (cont'd.)

- 1960s
  - Multiprogramming: many programs loaded and sharing the CPU
    - Passive multiprogramming: interrupts
    - Active multiprogramming: time slicing
- 1970s
  - Faster CPUs: caused problems with slower I/O devices
  - Main memory physical capacity limitations
    - Multiprogramming schemes: increase CPU use
    - Virtual memory system: solves physical limitation

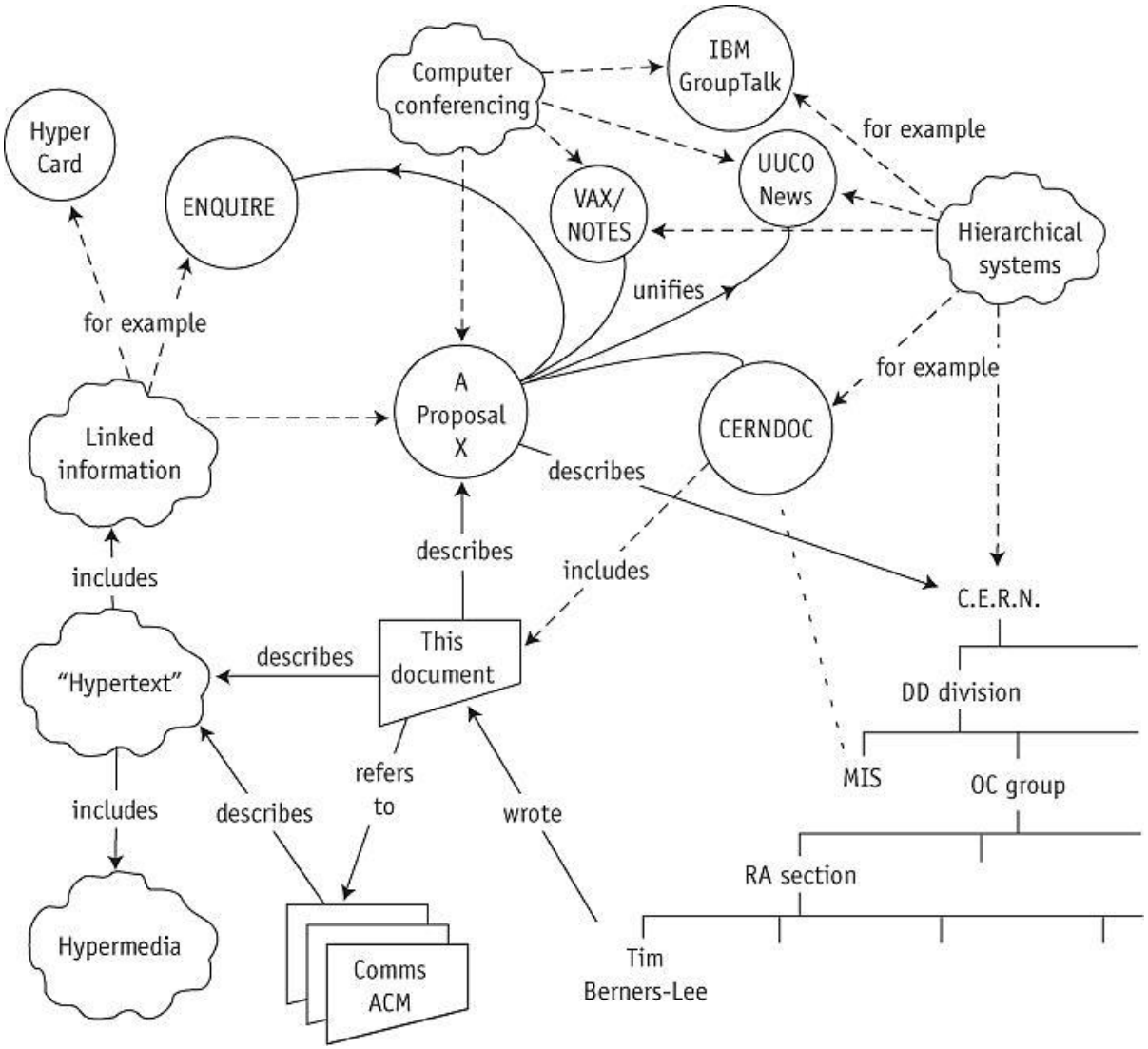
Brief History of Operating Systems Development (cont'd.)

- 1980s
  - More flexible hardware
    - Logical functions built on replaceable circuit boards
  - Operating system functions
    - Made part of the hardware (firmware)
  - Multiprocessing: more than one processor
    - Allowed parallel program execution
  - Evolution of personal computers and high-speed communications
    - Led to distributed processing/networked systems

Brief History of Operating Systems Development (cont'd.)

- 1990s
  - Demand for Internet capability
  - Increased networking
  - Increased tighter security demands
    - Protect system assets from Internet threats
  - Multimedia applications
    - Demanded additional power, flexibility, device compatibility for most operating systems, etc.

(figure 1.11) Illustration from the 1989 proposal by Tim Berners-Lee describing his revolutionary “linked information system.”



Brief History of Operating Systems Development (cont'd.)

- 2000s
  - Need for improved flexibility, reliability, and speed
  - Virtual machines
    - Accommodated multiple operating systems that run at the same time and share resources
    - Required OS to have an intermediate manager
      - Oversee the access of each operating system to the server's physical resources
  - Multicore processors: two to many cores
    - What capabilities does the operating system need for these processors?

Brief History of Operating Systems Development (cont'd.)

- 2010s
  - Increased mobility and wireless connectivity
    - Proliferation of dual-core, quad-core, and other multicore CPUs
  - Multicore technology
    - Single chip equipped with two or more processor cores
    - What is the advantage over chips with transistors in close proximity?

Design Considerations

- Most common overall goal
  - Maximize use of the system's resources (memory, processing, devices, and files) and minimize downtime
- Factors included in developmental efforts
  - RAM resources
  - CPUs: number and type available
  - Peripheral devices: variety likely to be connected
  - Networking capability
  - Security requirements, etc.

Conclusion

- Overall function of operating systems
- Evolution of operating systems
  - Capable of running complex computers and computer systems
- Operating system designer
  - Chooses the policies that best match the system's environment