Understanding Operating Systems
Fifth Edition

Chapter 1

Introducing Operating Systems

Learning Objectives

- 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 (continued)

- The differences among batch, interactive, real-time, hybrid, and embedded operating systems
- Multiprocessing and its impact on the evolution of operating system software
- Virtualization and core architecture trends in new operating systems

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 describes:
  - How operating systems work
  - The evolution of operation systems

What is an Operating System?

- **Computer System**
  - Software (programs)
  - Hardware (physical machine and electronic components)
- **Operating System**
  - Part of computer system (software)
  - Manages all hardware and software
    - Controls every file, device, section of main memory and nanosecond of processing time
    - Controls *who* can use the system
    - Controls *how* system is used

Operating System Software

- Includes four essential subsystem managers
  - Memory Manager
  - Processor Manager
  - Device Manager
  - File Manager
- Network Manager (fifth subsystem manager)
  - In all modern operating systems
  - Assumes responsibility for networking tasks
  - Discussed further in Chapters 9 & 10

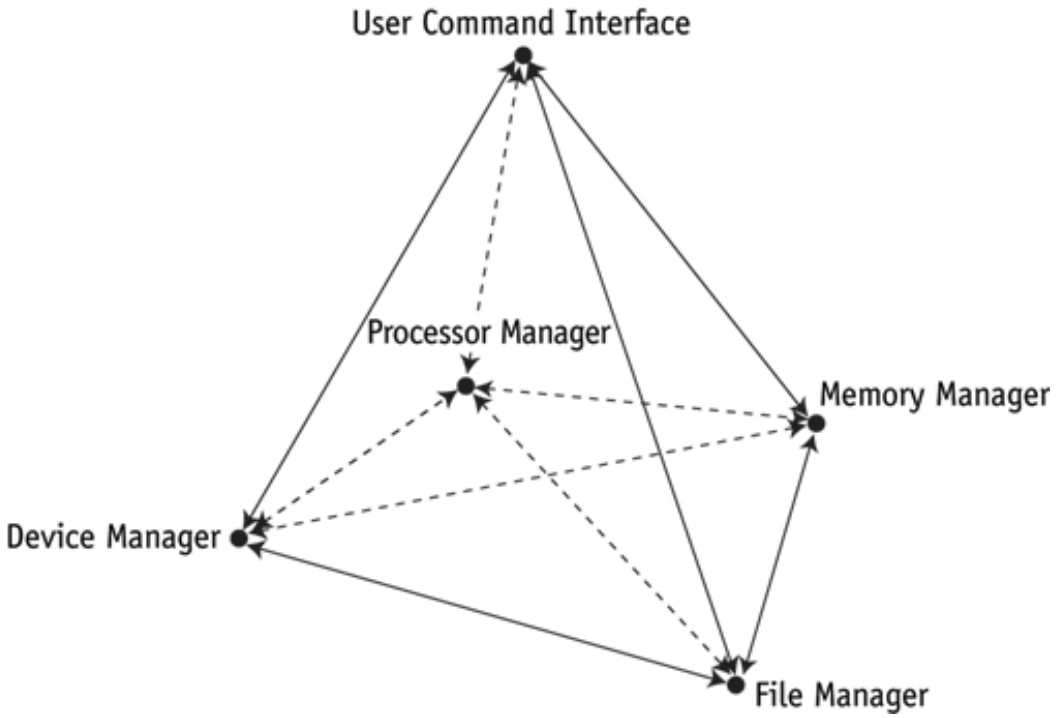
Operating System Software (continued)

- **User Command Interface**
  - Provides user communication
    - User issues *commands* to operating system
  - Unique to each operating system
    - May vary between versions
  - Essential managers provide support

Operating System Software (continued)

(figure 1.1)

This model of a non-networked operating system shows four subsystem managers supporting the User Command Interface.



Operating System Software (continued)

- Each manager:
  - Works closely with other managers
  - Performs a unique role
- Manager tasks
  - Monitor its resources continuously
  - Enforce policies determining:
    - Who gets what, when, and how much
  - Allocate the resource (when appropriate)
  - Deallocate the resource (when appropriate)

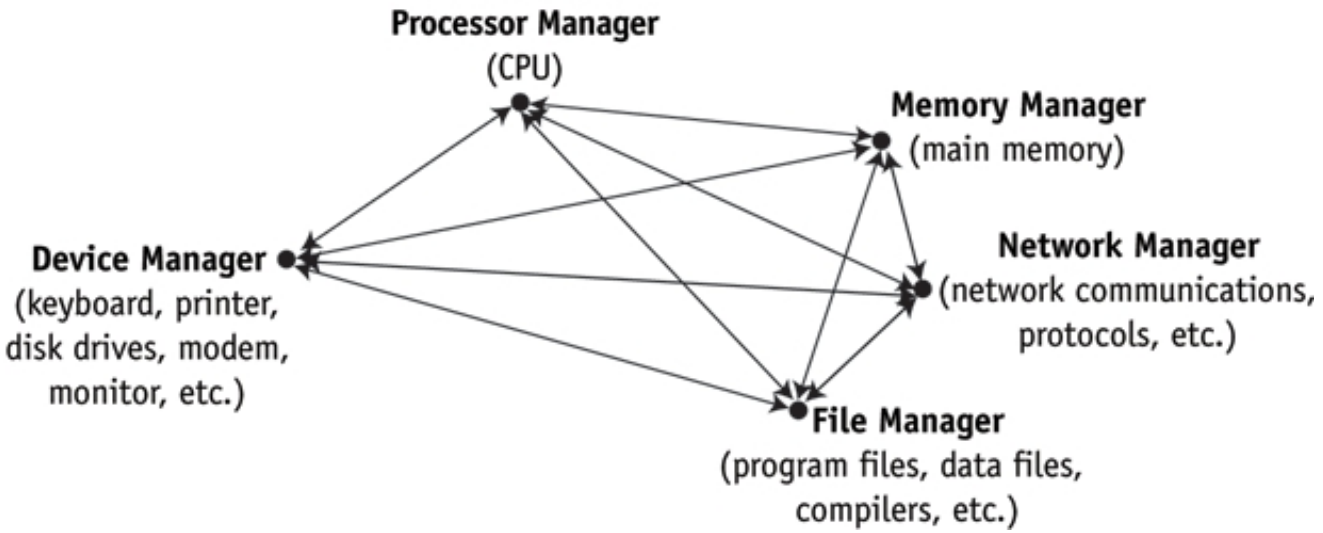
Operating System Software (continued)

- **Network Manager**
  - Operating systems with networking capability
  - Fifth essential manager
  - Convenient way for users to share resources
  - Retains user access control
- Resources include:
  - Hardware (CPUs, memory areas, printers, tape drives, modems, and disk drives)
  - Software (compilers, application programs, and data files)

Operating System Software (continued)

(figure 1.2)

Networked systems have a Network Manager that assumes responsibility for networking tasks while working harmoniously with every other manager.



Main Memory Management

- In charge of main memory
  - Random Access Memory (RAM)
- Responsibilities include:
  - Preserving space in main memory occupied by operating system
  - Checking validity and legality of memory space request
  - Setting up memory tracking table
    - Tracks usage of memory by sections
    - Needed in multiuser environment
  - Deallocating memory to reclaim it

Processor Management

- In charge of allocating **Central Processing Unit** (CPU)
- Tracks **process** status
  - An instance of program execution
- Two levels of responsibility:
  - Handle jobs as they enter the system
    - Handled by Job Scheduler
  - Manage each process within those jobs
    - Handled by Process Scheduler

Device Management

- In charge of monitoring all resources
  - Devices, channels, and control units
- Responsibilities include:
  - Choosing most efficient resource allocation method
    - Printers, ports, disk drives, etc.
    - Based on scheduling policy
  - Allocating the device
  - Starting device operation
  - Deallocating the device

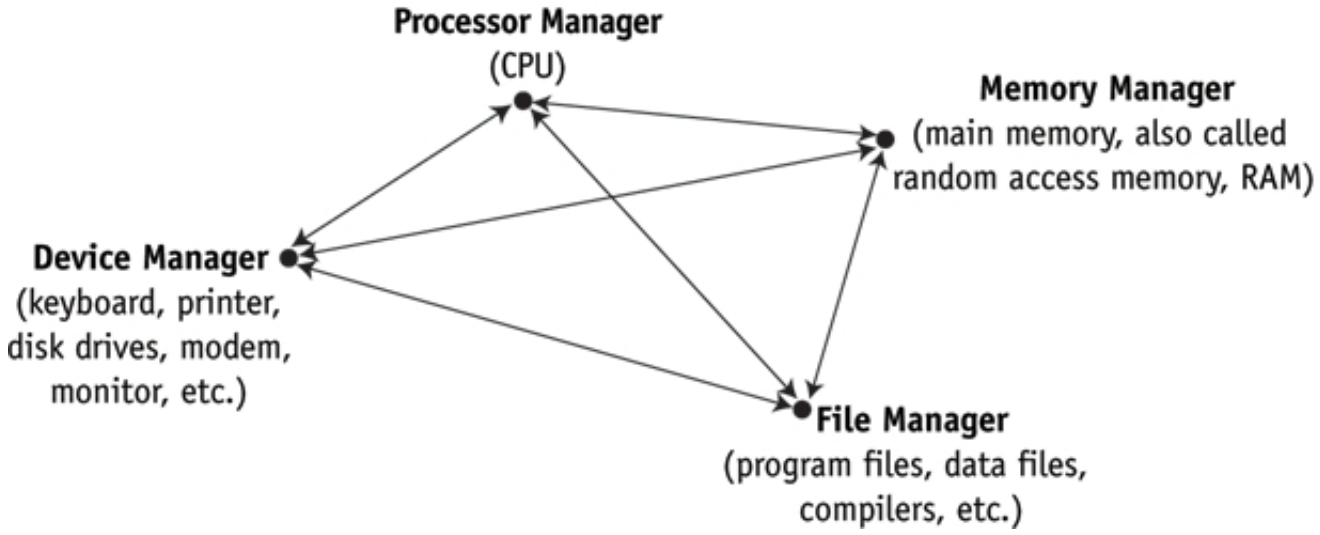
File Management

- In charge of tracking every file in the system
  - Data files, program files, compilers, application programs
- Responsibilities include:
  - Enforcing user/program resource access restrictions
    - Uses predetermined access policies
  - Controlling user/program modification restrictions
    - Read-only, read-write, create, delete
  - Allocating resource
    - Opening the file
    - Deallocating file (by closing it)

Cooperation Issues

- Essential manager
  - Perform individual tasks *and*
  - Harmoniously interact with other managers
    - Requires incredible precision
  - No single manager performs tasks in isolation
  - Network manager
    - Convenient way to share resources
    - Controls user access

Operating System Software (continued)



(figure 1.3)

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

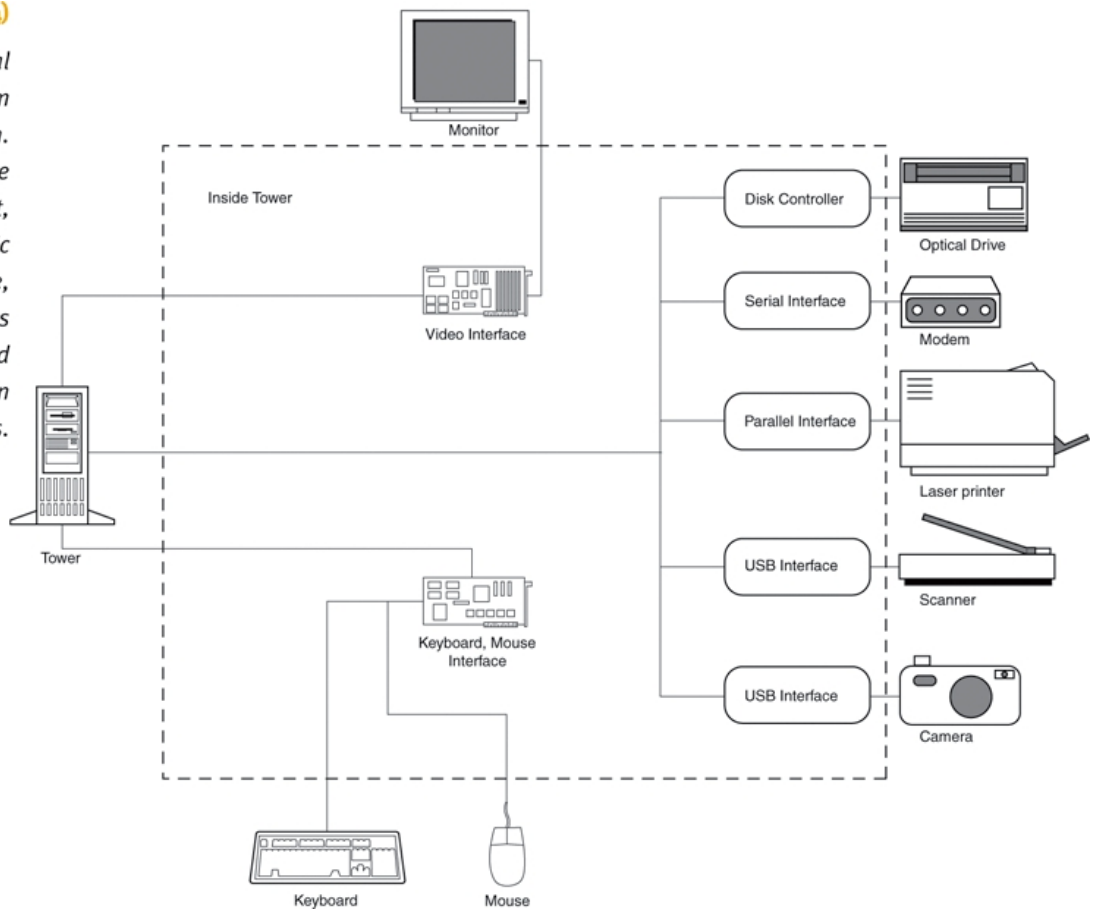
A Brief History of Machine Hardware

- **Hardware:** physical machine and electronic components
  - **Main memory (RAM)**
    - Data/Instruction storage and execution
  - **Input/Output devices (I/O devices)**
    - All peripheral devices in system
    - Printers, disk drives, CD/DVD drives, flash memory, and keyboards
  - **Central processing unit (CPU)**
    - Controls interpretation and execution of instructions
    - Controls operation of computer system

A Brief History of Machine Hardware (continued)

(figure 1.4)

A logical view of a typical computer system hardware configuration. The tower holds the Central Processing Unit, the Arithmetic and Logic Unit, registers, cache, and main memory, as well as controllers and interfaces shown within the dotted lines.



A Brief History of Machine Hardware (continued)

- Computer classification
  - By capacity and price (until mid-1970s)
- Mainframe
  - Large machine
    - Physical size and internal memory capacity
  - Classic Example: 1964 IBM 360 model 30
    - CPU required 18-square-foot air-conditioned room
    - CPU size: 5 feet high x 6 feet wide
    - Internal memory: 64K
    - Price: $200,000 (1964 dollars)
  - Applications limited to large computer centers

A Brief History of Machine Hardware (continued)

- **Minicomputer**
  - Developed for smaller institutions
  - Compared to mainframe
  - Smaller in size and memory capacity
    - Cheaper
  - Example: Digital Equipment Corp. minicomputer
    - Price: less than $18,000
  - Today
    - Known as **midrange computers**
    - Capacity between microcomputers and mainframes

A Brief History of Machine Hardware (continued)

- **Supercomputer**
  - Massive machine
  - Developed for military operations and weather forecasting
  - Example: Cray supercomputer
    - 6 to 1000 processors
    - Performs up to 2.4 trillion floating-point operations per second (teraflops)
  - Uses:
    - Scientific research
    - Customer support/product development

A Brief History of Machine Hardware (continued)

- **Microcomputer**
  - Developed for single users in the late 1970s
  - Example: microcomputers by Tandy Corporation and Apple Computer, Inc.
    - Very little memory (by today's standards)
    - 64K maximum capacity
  - Microcomputer's distinguishing characteristic
    - Single-user status

A Brief History of Machine Hardware (continued)

- **Workstations**
  - Most powerful microcomputers
  - Developed for commercial, educational, and government enterprises
  - Networked together
  - Support engineering and technical users
    - Massive mathematical computations
    - Computer-aided design (CAD)
  - Applications
    - Requiring powerful CPUs, large main memory, and extremely high-resolution graphic displays

A Brief History of Machine Hardware (continued)

- **Servers**
  - Provide specialized services
    - To other computers or client/server networks
  - Perform critical network task
  - Examples:
    - Print servers
    - Internet servers
    - Mail servers

A Brief History of Machine Hardware (continued)

- **Advances in computer technology**
  - Dramatic changes
    - Physical size, cost, and memory capacity
  - Networking
    - Integral part of modern computer systems
  - Mobile society information delivery
    - Creating strong market for handheld devices
  - New classification
    - By processor capacity, not memory capacity
  - Moore's Law
    - Computing power rises exponentially

A Brief History of Machine Hardware (continued)

Platform	Operating System
Microcomputers	Linux, UNIX, Windows
Minicomputers	Linux, IBM OS/400, OpenVMS, UNIX
Mainframe computers	IBM OS/390, Linux, UNIX
Supercomputers	IRIX, UNICOS
Workstations, Servers	Linux, UNIX, Windows
Networks	Linux, NetWare, UNIX, Windows
Personal Digital Assistants	Palm OS, Windows Mobile

(table1.1)

A brief list of platforms and sample operating systems (listed here in alphabetical order).

Types of Operating Systems

- Five categories
  - Batch
  - Interactive
  - Real-time
  - Hybrid
  - Embedded
- Two distinguishing features
  - Response time
  - How data enters into the system

Types of Operating Systems (continued)

- **Batch Systems**
  - Input relied on punched cards or tape
  - Efficiency measured in throughput
- **Interactive Systems**
  - Faster turnaround than batch systems
  - Slower than real-time systems
  - Introduced to provide fast turnaround when debugging programs
  - Time-sharing software developed for operating system

Types of Operating Systems (continued)

- **Real-time systems**
  - Reliability is key
  - Fast and time limit sensitive
  - Used in time-critical environments
    - Space flights, airport traffic control, high-speed aircraft
    - Industrial processes
    - Sophisticated medical equipment
    - Distribution of electricity
    - Telephone switching
  - Must be 100% responsive, 100% of the time

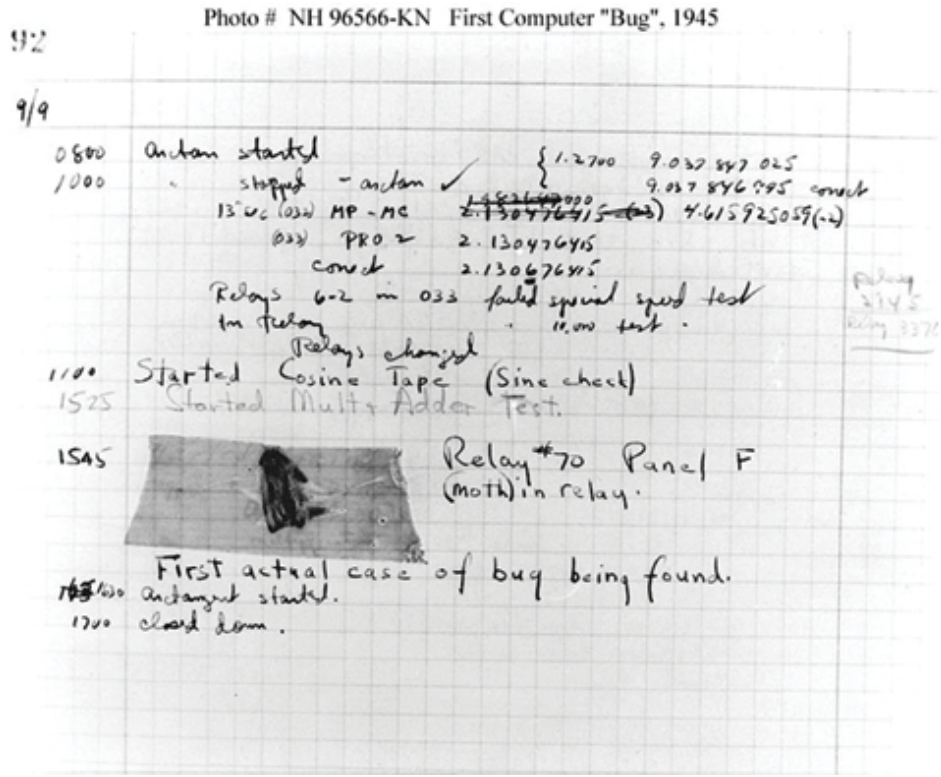
Types of Operating Systems (continued)

- **Hybrid systems**
  - Combination of batch and interactive
  - Accept and run batch programs in the background
    - Interactive load is light
- **Embedded systems**
  - Computers placed inside other products
  - Adds features and capabilities
  - Operating system requirements
    - Perform specific set of programs
    - Not interchangeable among systems
    - Small kernel and flexible function capabilities

Brief History of Operating Systems Development

- **1940s: first generation**
  - Computers based on vacuum tube technology
  - No standard operating system software
  - Typical program included every instruction needed by the computer to perform the tasks requested
  - Poor machine utilization
    - CPU processed data and performed calculations for fraction of available time
  - Early programs
    - Designed to use the resources conservatively
    - Understandability is not a priority

Brief History of Operating Systems Development (continued)



(figure 1.7)

Dr. Grace Hopper's research journal from her work on Harvard's Mark I computer in 1945 included the remains of the first computer "bug," a moth that had become trapped in the computer's relays causing the system to crash. The term is still in use today.

Brief History of Operating Systems Development (continued)

- **1950s: second generation**
  - Focused on cost effectiveness
  - Computers were expensive
    - IBM 7094: $200,000
  - Two widely adopted improvements
    - Computer operators: humans hired to facilitate machine operation
    - Concept of job scheduling: group together programs with similar requirements
  - Expensive time lags between CPU and I/O devices

Brief History of Operating Systems Development (continued)

- **1950s: second generation (continued)**
  - I/O device speed gradually became faster
    - Tape drives, disks, and drums
  - Records blocked *before* retrieval or storage
  - Access methods developed
    - Added to object code by linkage editor
  - Buffer between I/O and CPU introduced
    - Reduced speed discrepancy
  - Timer interrupts developed
    - Allowed job-sharing

Brief History of Operating Systems Development (continued)

- **1960s: third generation**
  - Faster CPUs
  - Speed caused problems with slower I/O devices
  - Multiprogramming
    - Allowed loading many programs at one time
  - Program scheduling
    - Initiated with second-generation systems
    - Continues today
  - Few advances in data management
  - Total operating system customization
    - Suit user's needs

Brief History of Operating Systems Development (continued)

- **1970s**
  - Faster CPUs
  - Speed caused problems with slower I/O devices
  - Main memory physical capacity limitations
    - Multiprogramming schemes used to increase CPU
    - Virtual memory developed to solve physical limitation
  - Database management software
    - Became a popular tool
  - A number of query systems introduced
  - Programs started using English-like words, modular structures, and standard operations

Brief History of Operating Systems Development (continued)

(figure 1.9)

The Cray I supercomputer, introduced in 1976, boasted 8 MB main memory and a world-record speed of 160 million floating-point operations per second. Its circular design meant that no wire was more than 4 feet (1.2 meters) long.

Brief History of Operating Systems Development (continued)

- **1980s**
  - **Cost/performance ratio** improvement of computer components
  - More flexible hardware (firmware)
  - **Multiprocessing**
    - Allowed parallel program execution
  - Evolution of personal computers
  - Evolution of high-speed communications
  - **Distributed processing** and **networked systems** introduced

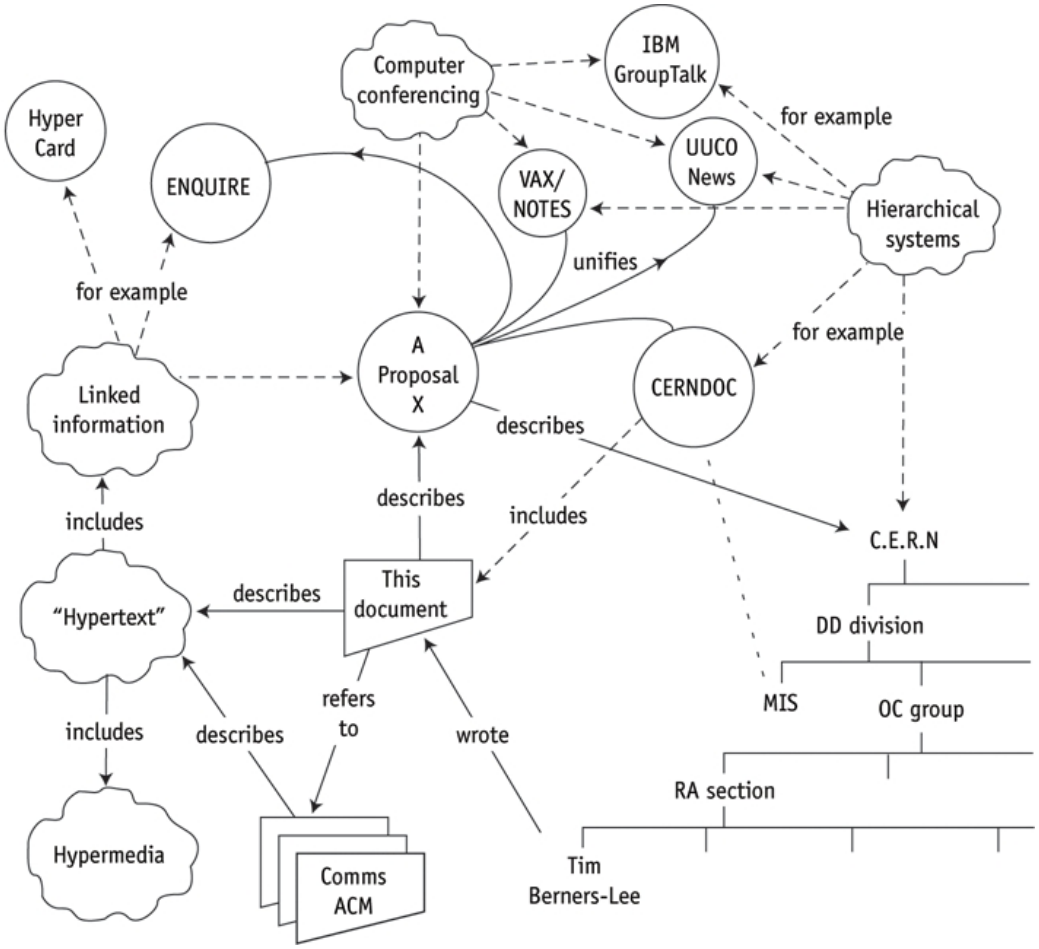
Brief History of Operating Systems Development (continued)

- **1990s**
  - Demand for Internet capability
    - Sparked proliferation of networking capability
    - Increased networking
    - Increased tighter security demands to protect hardware and software
  - Multimedia applications
    - Demanding additional power, flexibility, and device compatibility for most operating systems

Brief History of Operating Systems Development (continued)

(figure 1.10)

Illustration from the first page of the 1989 proposal by Tim Berners-Lee describing his revolutionary "linked information system." Based on this research, he designed the first World Wide Web server and browser, making it available to the general public in 1991.



Brief History of Operating Systems Development (continued)

- **2000s**
  - Primary design features support:
    - Multimedia applications
    - Internet and Web access
    - Client/server computing
  - Computer systems requirements
    - Increased CPU speed
    - High-speed network attachments
    - Increased number and variety of storage devices
  - Virtualization
    - Single server supports different operating systems

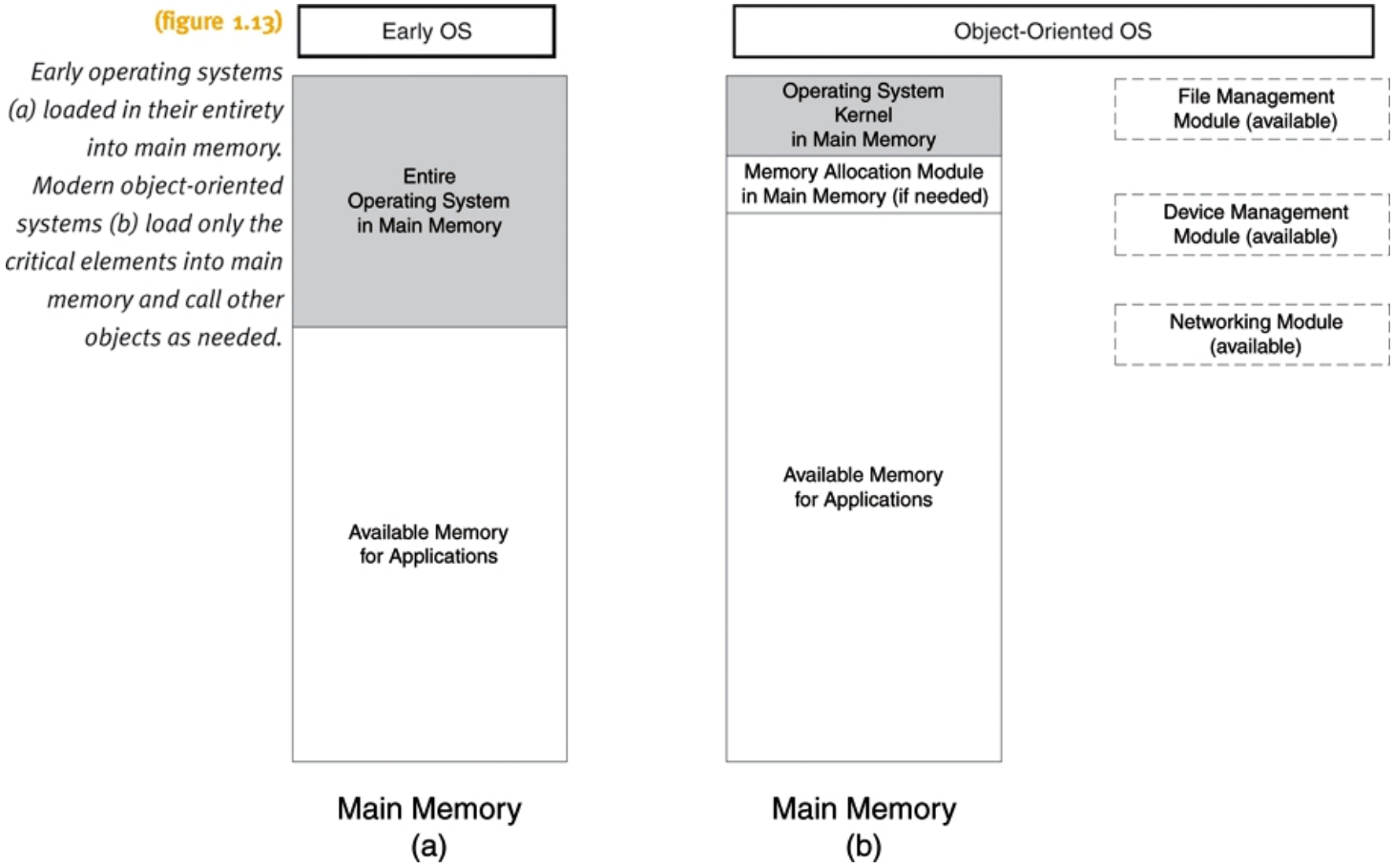
Threads

- Multiple actions executing simultaneously
  - Heavyweight process (conventional process)
    - Owns the resources
    - Passive element
  - Lightweight process (thread)
    - Uses CPU and scheduled for execution
    - Active element
  - Multithreaded applications programs
    - Contain several threads running at one time
    - Same or different priorities
    - Examples: Web browsers and time-sharing systems

Object-Oriented Design

- Driving force in system architecture improvements
  - **Kernel** (operating system nucleus)
    - Resides in memory at all times, performs essential tasks, and protected by hardware
  - **Kernel reorganization**
    - Memory resident: process scheduling and memory allocation
    - Modules: all other functions
  - Advantages
    - Modification and customization without disrupting integrity of the remainder of the system
    - Software development more productive

Object-Oriented Design (continued)



Summary

- Operating system overview
- Functions of OS
  - Manages computer system
    - Hardware and software
  - Four essential managers
    - Work closely with the other managers and perform unique role
  - Network Manager
    - Operating systems with networking capability
  - Essential hardware components
    - Memory chips, I/O, storage devices, and CPU

Summary (continued)

- Evolution of OSs
  - Run increasingly complex computers
  - Run increasingly complex computer systems
  - Prior to mid-1970s
    - Computers classified by capacity and price
  - Dramatic changes over time
    - Moore's Law: computing power rises exponentially
    - Physical size, cost, and memory capacity
- Mobile society information delivery
  - Creates strong market for handheld devices
  - Integral in modern computer systems

Summary (continued)

- Five categories of operating systems
  - Batch, interactive, real-time, hybrid, and embedded
- Use of object-oriented design improves the system architecture
- Several ways to perform OS tasks
- Designer determines policies to match system's environment
- Next:
  - Explore details of operating system components