Chapter 1

Introducing Operating Systems

Understanding Operating Systems, Fourth Edition

Objectives

You will be able to describe:

- 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, and give at least one example of an operating system for each of the following: PDAs, microcomputers, minicomputers, mainframes, workstations, and supercomputers

Objectives (continued)

Describe:

- The differences between the following types of operating systems: batch, interactive, real-time, hybrid, and embedded
- Multiprocessing and its impact on the evolution of operating system software
- System architecture trends in current operating systems

Understanding Operating Systems

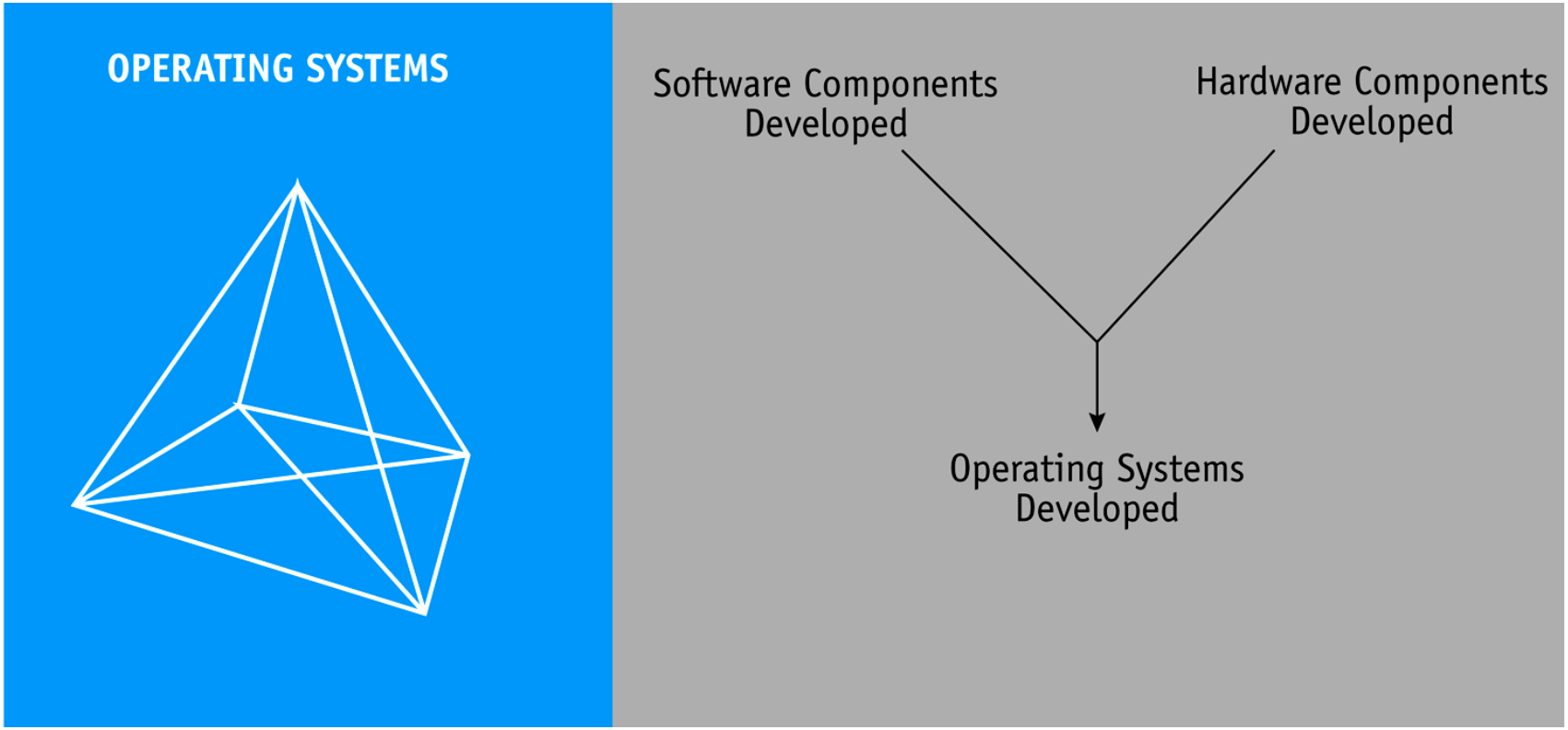
1943 quote

“I think there is a world market
for maybe five computers.”

Thomas J. Watson (1874–1956)

- chairman of IBM 1949–1956

Understanding Operating Systems



What is an Operating System?

- **Operating System:**
  - Part of the computing system that manages all of the hardware and software
  - Controls every file, every device, every section of main memory, and every nanosecond of processing time
  - Controls who can use the system and how

What is an Computer System?

Computer system consists of:

- Software (programs)
- Hardware (the physical machine and its electronic components)

And also:

- people
- processes
- procedures

Operating System Software

- **Essential managers** of an operating system:
  - Memory Manager
  - Processor Manager
  - Device Manager
  - File Manager
- Each manager both works closely with the other managers and performs its unique role
- User Command Interface is unique to each operating system

Operating System Software (continued)

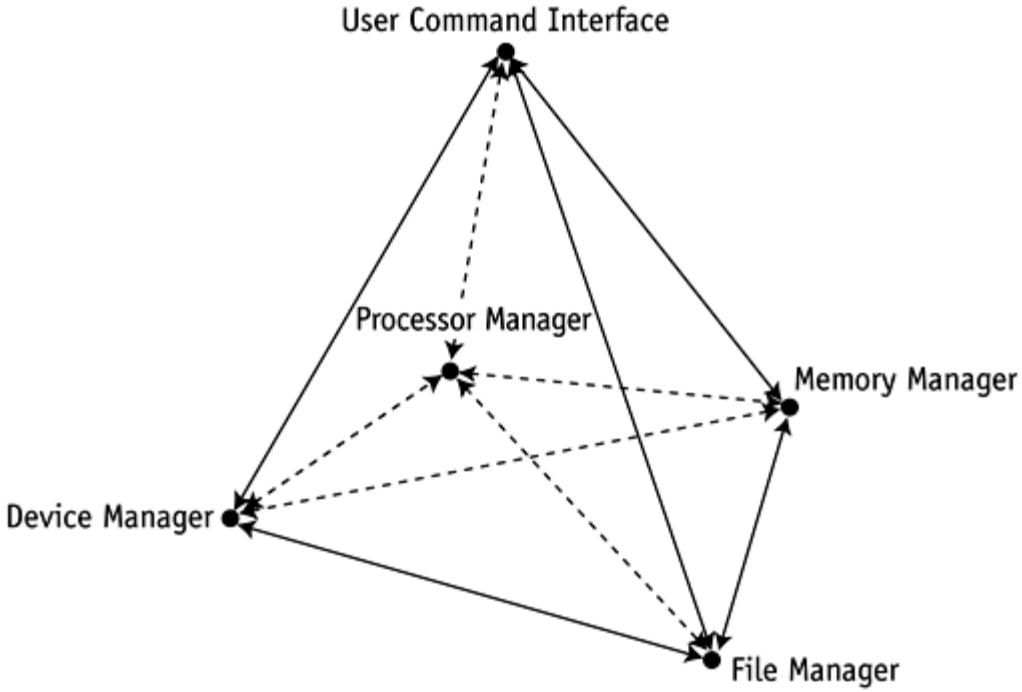
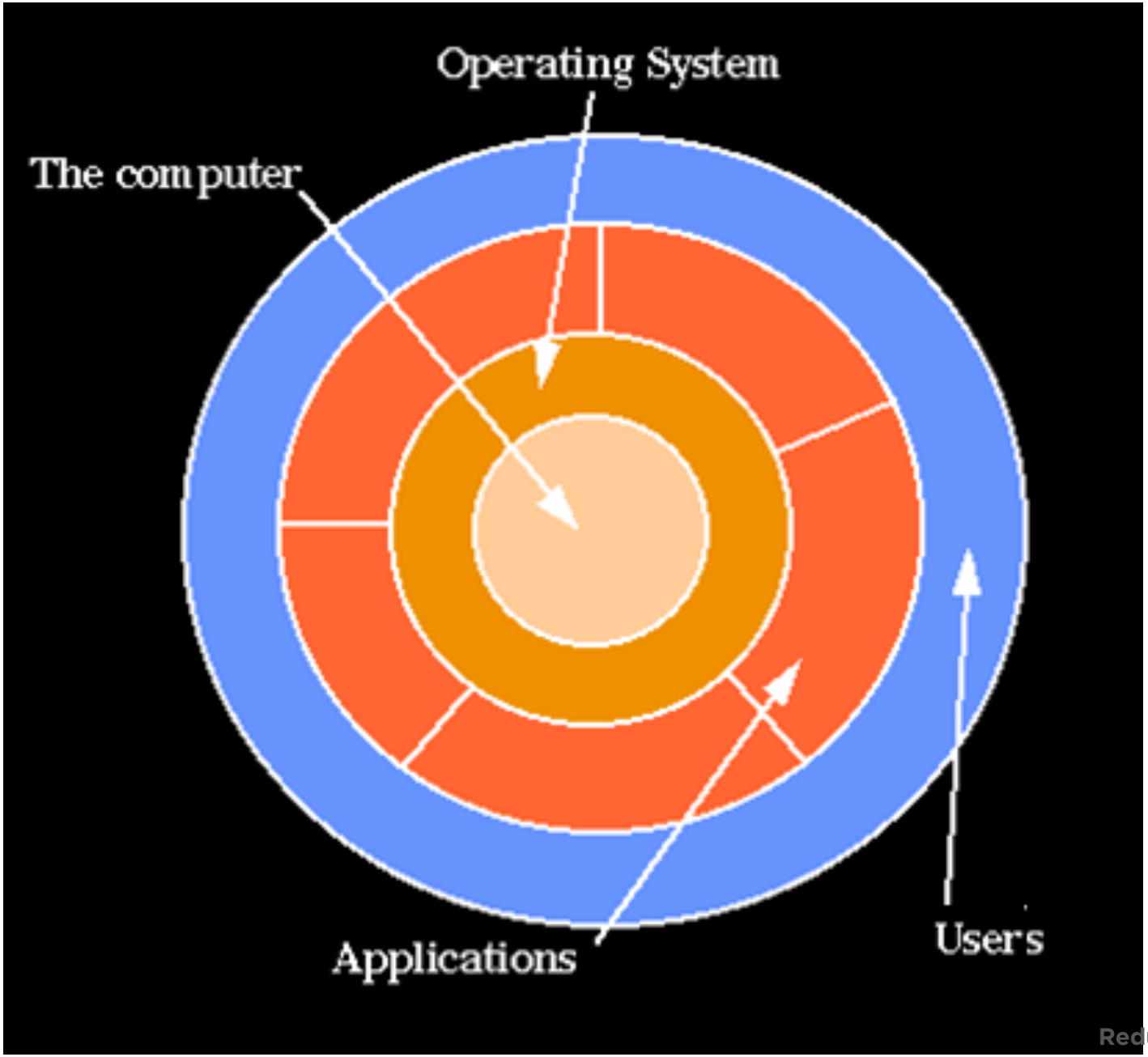


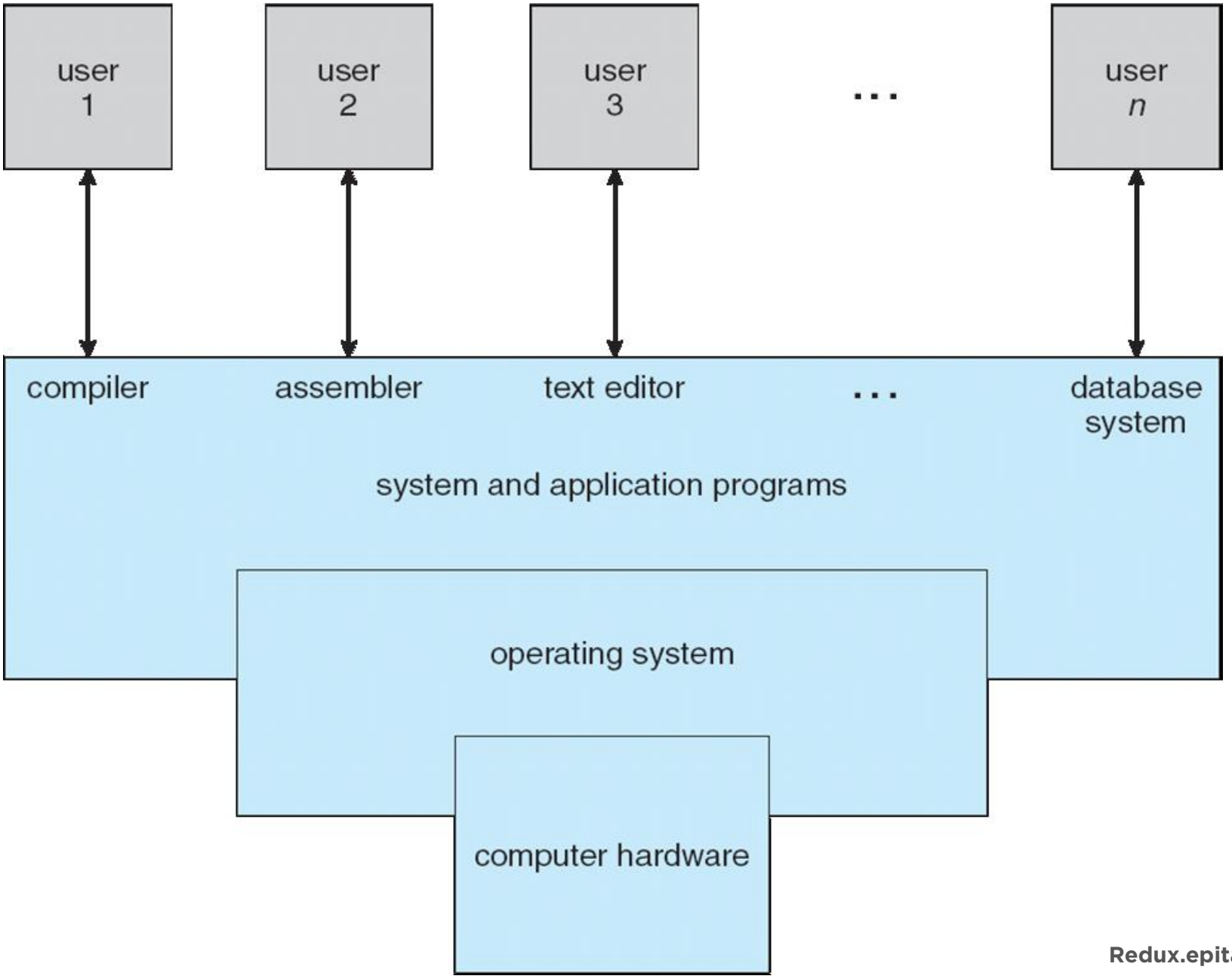
Figure 1.1: Model of a non-networked operating system

Operating System Software (continued)

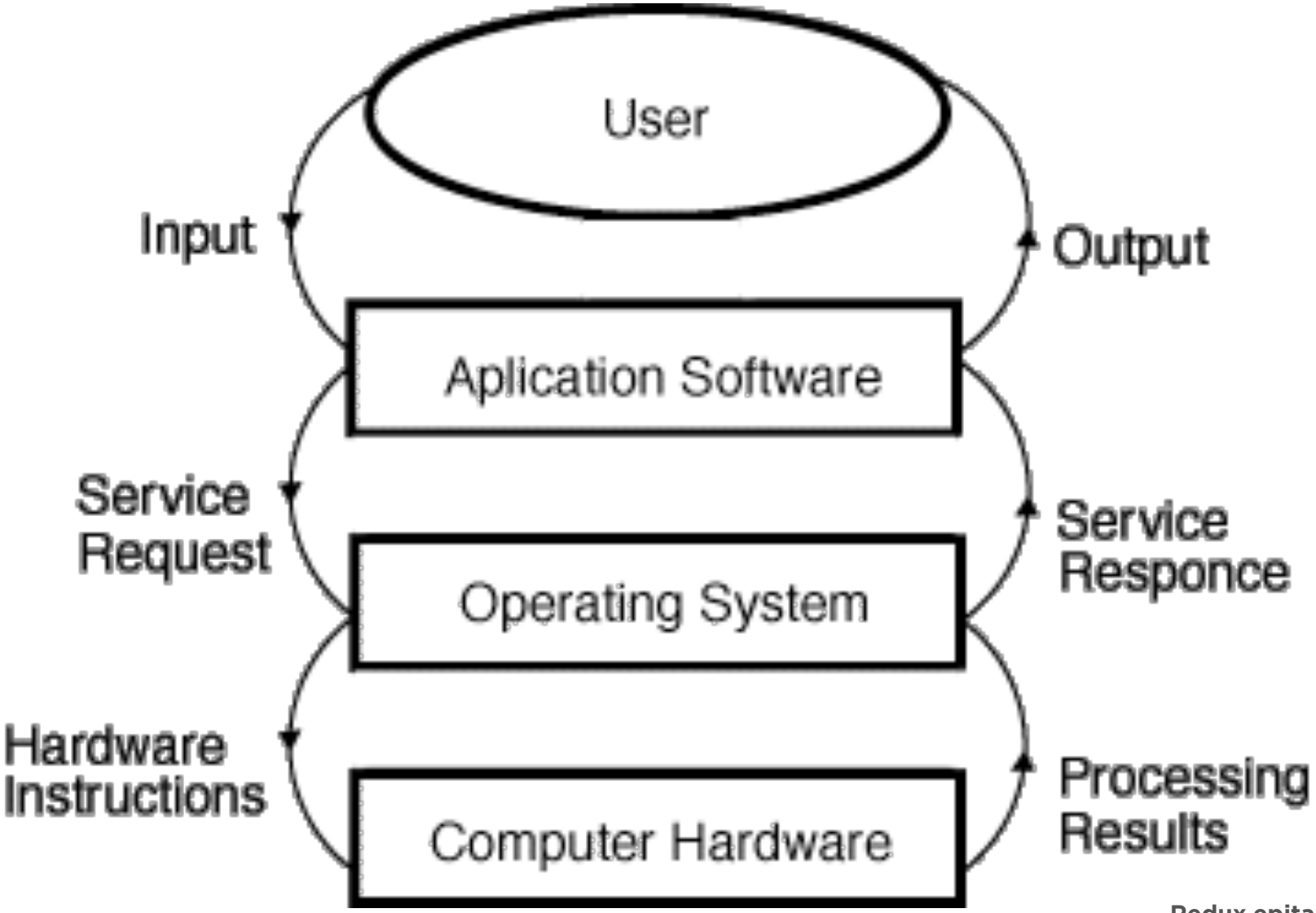
- Each **subsystem manager must perform** the following tasks:
  - Monitor its resources continuously
  - Enforce the policies that determine who gets what, when, and how much
  - Allocate the resource when it's appropriate
  - Deallocate the resource when appropriate



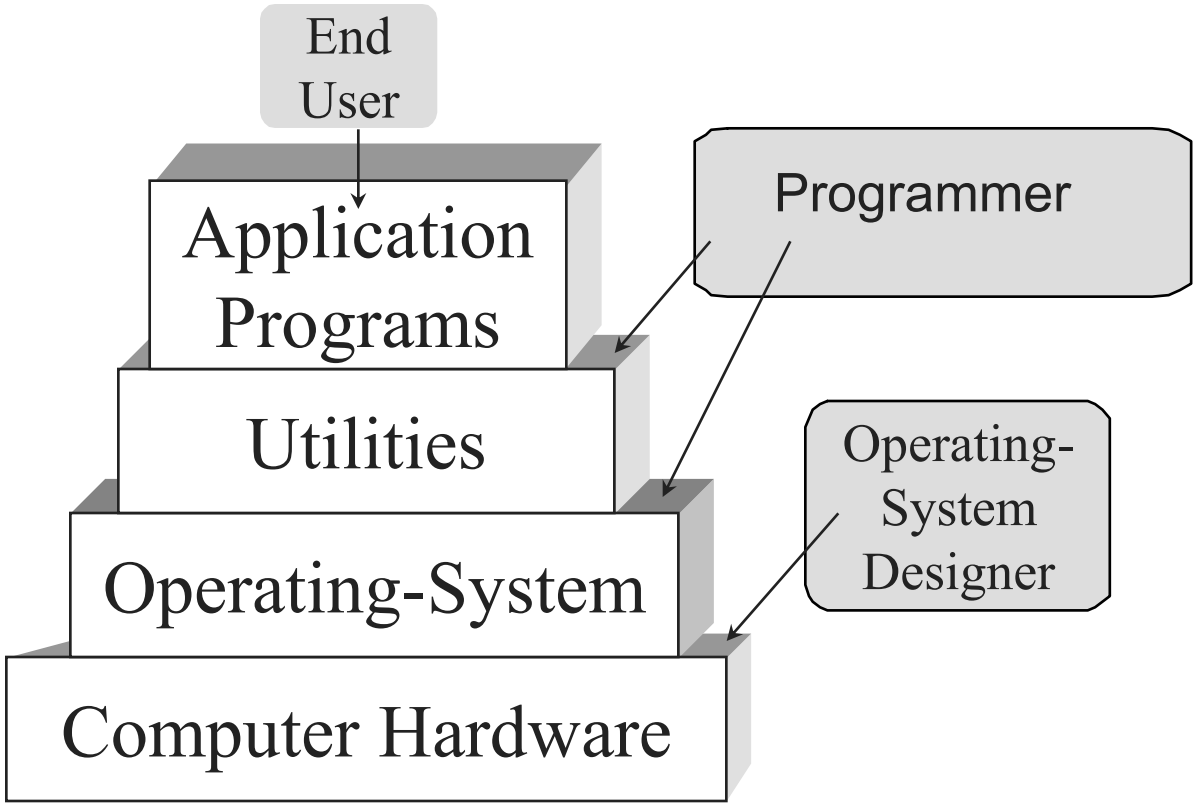
Redux.epita.cf



Redux.epita.cf



Redux.epita.cf


End User
Application Programs
Utilities
Operating-System
Computer Hardware
Programmer
Operating-System Designer

Operating System Software (continued)

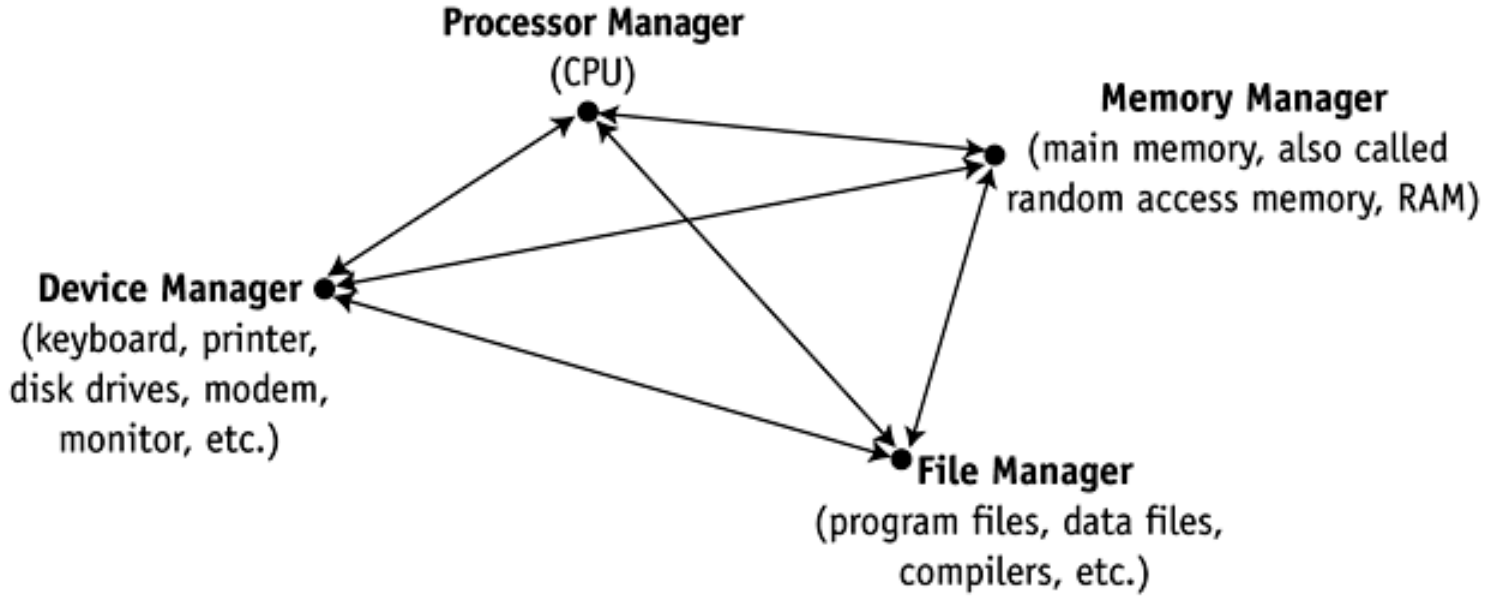


Figure 1.2: Subsystems managers at the base of a pyramid

Operating System Software (continued)

- **Memory Manager:** In charge of main memory (RAM)
- **Responsibilities include:**
  - Preserves the space in main memory occupied by the operating system
  - Checks the validity of each request for memory space
  - Sets up a table to keep track of who is using which section of memory in a multiuser environment
  - Deallocates memory when the time comes to reclaim the memory

Operating System Software (continued)

- **Processor Manager** decides how to allocate the central processing unit (CPU)
- Processor Manager has **two levels of responsibility:**
  - To handle jobs as they enter the system
    - Handled by Job Scheduler
  - To manage each process within those jobs
    - Handled by Process Scheduler

Operating System Software (continued)

- **Device Manager** monitors every device, channel, and control unit
- **Responsibilities include:**
  - Chooses the most efficient way to allocate all of the system's devices, printers, terminals, disk drives, based on a scheduling policy
  - Makes the allocation, starts its operation
  - Deallocates the device

Operating System Software (continued)

- **File Manager** keeps track of every file in the system including data files, assemblers, compilers, and application programs
- **Responsibilities include:**
  - Enforces restrictions on who has access to which files by using predetermined access
  - Controls what users are allowed to do with files once they access them
  - Allocates the resource by opening the file and deallocates it by closing the file

Operating System Software (continued)

- Operating systems with networking capability have a fifth essential manager called the **Network Manager**
- Network Manager provides a convenient way for users to share resources while controlling users' access to them. The resources include:
  - Hardware (such as CPUs, memory areas, printers, tape drives, modems, and disk drives)
  - Software (such as compilers, application programs, and data files)

Operating System Software (continued)

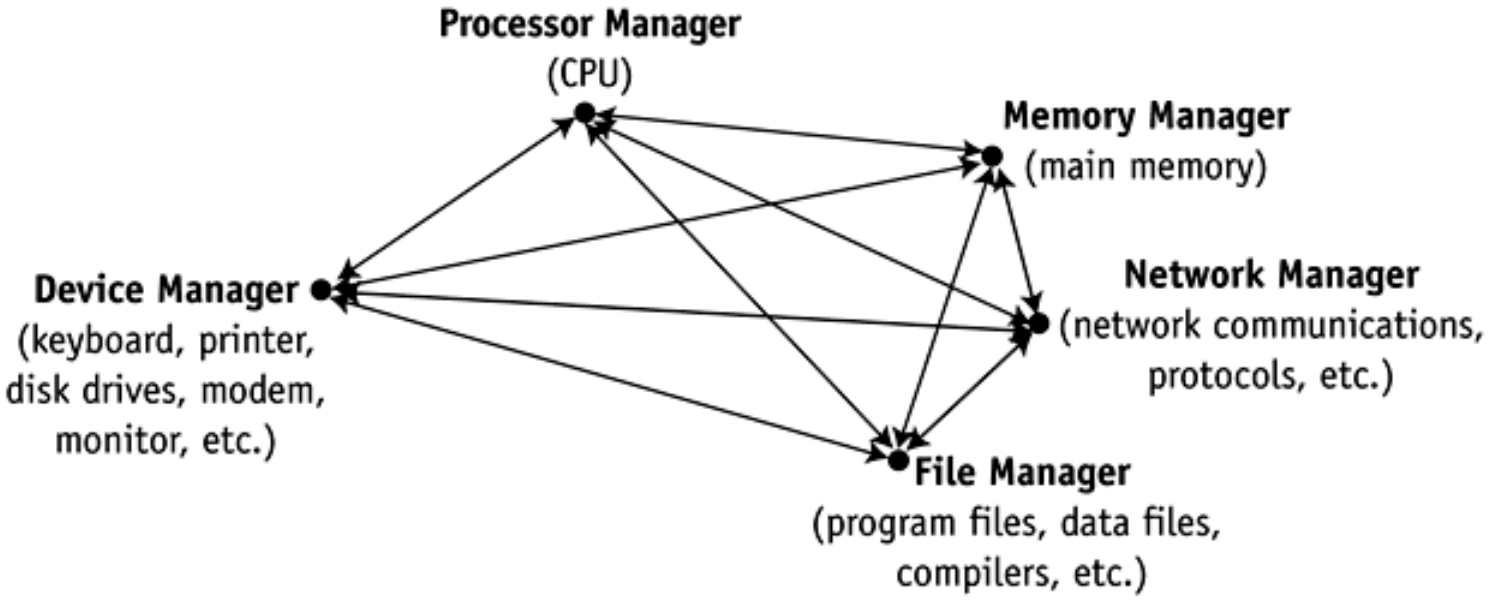


Figure 1.3: Model of a networked operating system

Redux.epita.cf

Machine Hardware

- Essential hardware components include:
  - Memory chips
  - Input/output devices
  - Storage devices
  - Central processing unit (CPU)

Machine Hardware (continued)

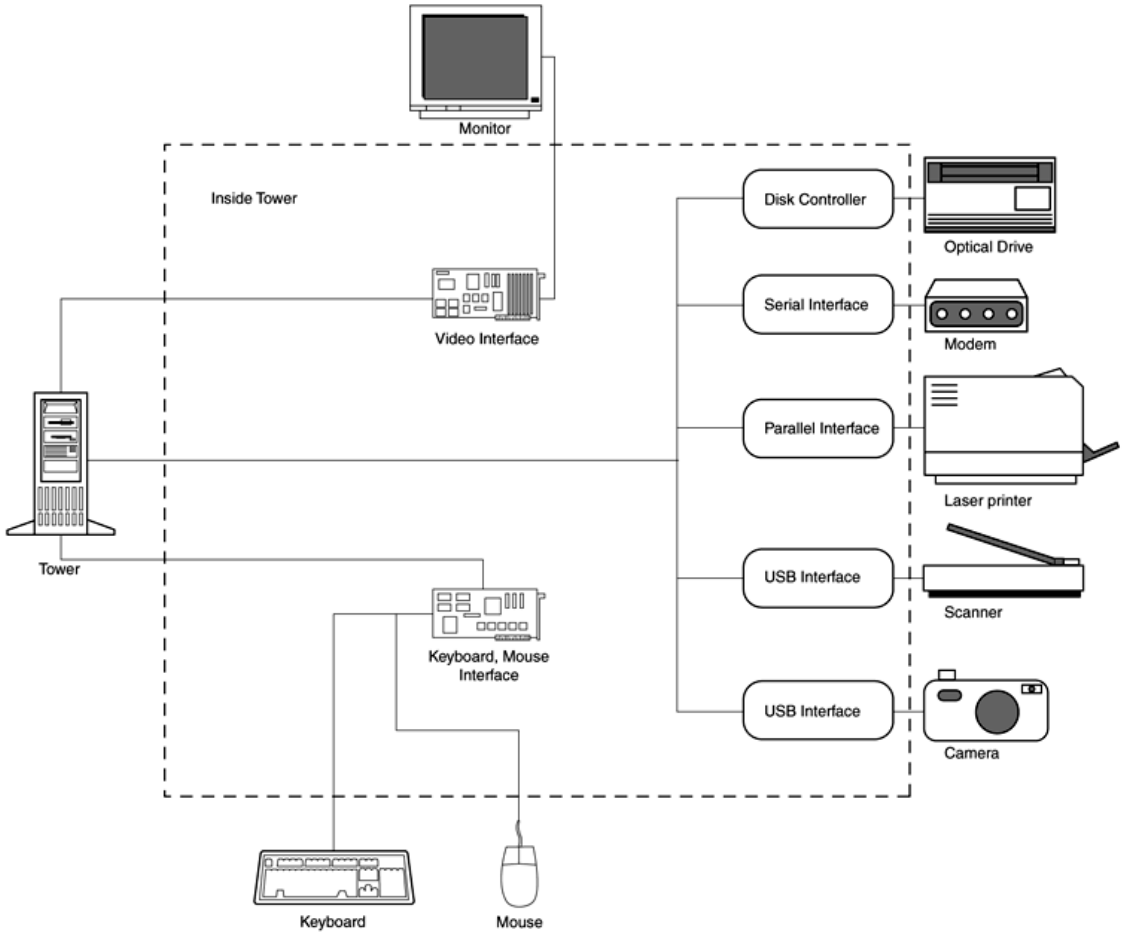


Figure 1.4: Computer system hardware configuration

Redux.epita.cf

Machine Hardware (continued)

- Until mid-1970s, computers were classified by capacity and price
- A **mainframe** was a large machine—in size and in internal memory capacity
  - In 1964, IBM 360 model 30 required an air-conditioned room (18 feet square) to house the CPU
  - The CPU was five feet high and six feet wide, had an internal memory of 64K
  - A price tag of $200,000 in 1964 dollars.
  - Applications limited to large computer centers

Machine Hardware (continued)

- **Minicomputer** was developed to meet the needs of smaller institutions
- Digital Equipment Corporation marketed one of the early minicomputers
- Price was less than $18,000
- Minicomputers are smaller in size and memory capacity, and cheaper than mainframes.
- Today, computers that fall between microcomputers and mainframes in capacity are often called midrange computers

Machine Hardware (continued)

- **Supercomputer** was introduced for military operations and weather forecasting
  - Example: A Cray supercomputer with six to thousands of processors performing up to 2.4 trillion floating point operations per second (teraflops)
- Supercomputer's uses include wide range of tasks from scientific research to customer support and product development

Machine Hardware (continued)

- **Microcomputer** was developed for single users in the late 1970s
- Tandy Corporation and Apple Computer, Inc. were the first to offer microcomputers
- These early models had very little memory by today's standards—64K maximum capacity
- The distinguishing characteristic of a microcomputer is its single-user status

Machine Hardware (continued)

- **Workstations:** Most powerful microcomputers used by commercial, educational, and government enterprises
- Workstations are networked together and used to support engineering and technical users who perform:
  - Massive mathematical computations
  - Computer-aided design (CAD)
  - Applications requiring powerful CPUs, large main memory, and extremely high-resolution graphic displays

Machine Hardware (continued)

- **Advances in computer technology**
  - Dramatic changes in physical size, cost, and memory capacity
  - Networking is an integral part of modern computer systems
  - Delivering information to a mobile society, creating a strong market for handheld devices
  - Classified by processor capacity instead of memory capacity
  - Computing power rises exponentially — Moore's Law

Machine Hardware (continued)

Platform	Operating System
Microcomputers	Linux, Macintosh, MS-DOS, Windows 2000/XP
Minicomputers	Linux, IBM OS/400, OpenVMS Alpha, UNIX
Mainframe computers	IBM OS/390, UNIX
Supercomputers	IRIX, UNICOS
Workstations	HP-UX, Sun Solaris, UNIX
Networks	Linux, NetWare, UNIX, Windows Server 2003
PDA	Palm OS, Windows CE

Table 1.1: Different platforms and operating systems

Types of Operating Systems (continued)

- Operating systems for computers fall into following five **categories**:
  - Batch
  - Interactive
  - Real-time
  - Hybrid
  - Embedded
- Distinguished by response time and how data is entered into the system

Types of Operating Systems (continued)

- **Batch Systems:**
  - Relied on punched cards or tape for input in past
  - Efficiency of the system was measured in throughput
- **Interactive Systems:**
  - Gives a faster turnaround than batch systems but are slower than the real-time systems
  - Introduced for users who needed fast turnaround when debugging their programs
  - Operating system required the development of time sharing software

Types of Operating Systems (continued)

- **Real-time systems:**
  - Fastest and used in time-critical environments
  - Real-time systems are used for:
    - Space flights, airport traffic control, high-speed aircraft
    - Industrial processes
    - Sophisticated medical equipment
    - Distribution of electricity
    - Telephone switching
  - A real-time system must be 100 percent responsive, 100 percent of the time

Types of Operating Systems (continued)

Figure 1.5: Computer interface box for the Apollo spacecraft (1968)

Types of Operating Systems (continued)

- **Hybrid Systems:**
  - Combination of batch and interactive
  - Accepts and runs batch programs in the background when the interactive load is light
- **Embedded Systems:**
  - Computers placed inside other products to add features and capabilities
  - Operating systems with small kernel and flexible functions capabilities will have potential for embedded system

Brief History of Operating Systems Development

- **1940s:**
  - 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
  - Machines were poorly utilized
    - CPU processed data and made calculations for only a fraction of the available time
  - Early programs were designed to use the resources conservatively at the expense of understandability

Redux.epita.cf

Brief History of Operating Systems Development (continued)

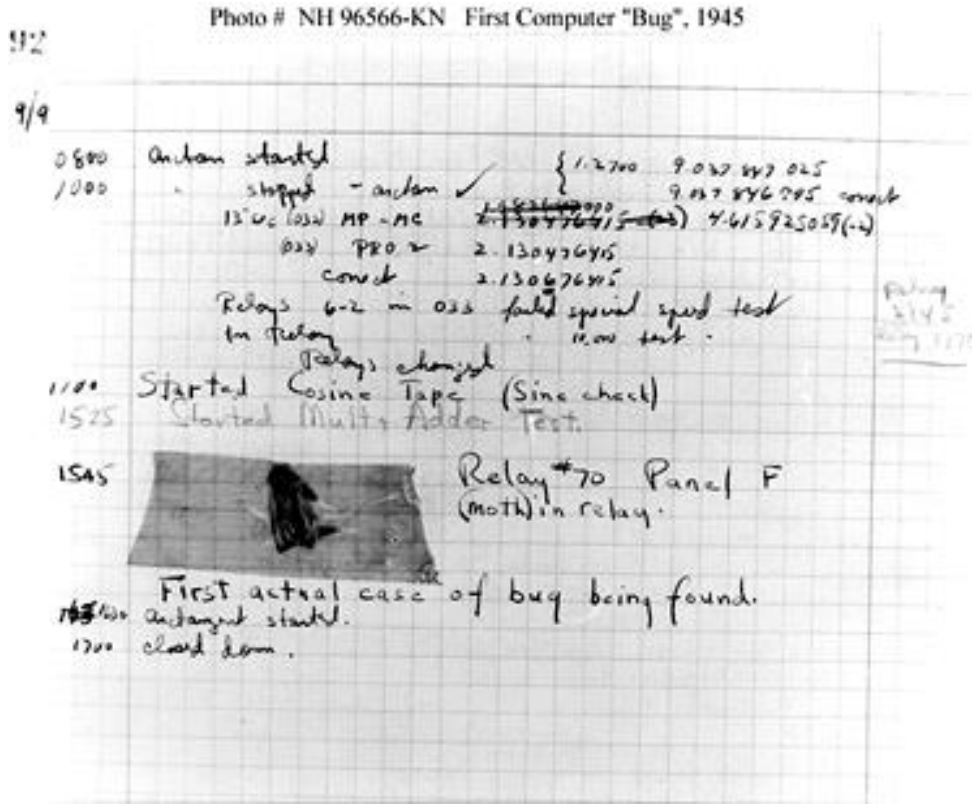


Figure 1.6: Remains of the first computer “bug,” a moth

Redux.epita.cf

Brief History of Operating Systems Development (continued)

- **1950s:**
  - Placed importance on cost effectiveness
  - Computers were still very expensive
    - IBM 7094 was priced at $200,000
  - Two improvements were widely adopted
    - Computer operators were hired to facilitate each machine's operation
    - Concept of job scheduling—groups together programs with similar requirements
  - Expensive time lags between CPU and I/O devices

Brief History of Operating Systems Development (continued)

Figure 1.7: The IBM 650 Magnetic Drum Data Processing System Machine

Redux.epita.cf

Brief History of Operating Systems Development (continued)

- Factors that improved the performance of CPU:
  - Speed of I/O devices like tape drives, disks, and drums gradually became faster
  - Records were blocked before they were retrieved or stored
  - Access methods were developed and added to object code by the linkage editor
  - Buffer was introduced between I/O and the CPU to reduce the discrepancy in speed
  - Timer interrupts were developed to allow job-sharing

Brief History of Operating Systems Development (continued)

- **1960s:**
  - Faster CPUs, but their speed caused problems
  - Multiprogramming was introduced, which allowed loading many programs at one time
  - Program scheduling, which was begun with second-generation systems, continued at this time
  - Few advances were made in data management
  - Total operating system was customized to suit user's needs

Brief History of Operating Systems Development (continued)

- **1970s:**
  - Faster CPUs, but their speed caused problems
  - Multiprogramming schemes to increase CPU use were limited by physical capacity of main memory
    - Development of virtual memory to solve physical limitation issue
  - Database management software became a popular tool
  - A number of query systems were introduced
  - Programs started using English-like words, modular structures, and standard operations

Brief History of Operating Systems Development (continued)

Figure 1.8: Cray I supercomputer, introduced in 1976

Redux.epita.cf

Brief History of Operating Systems Development (continued)

- **1980s:**
  - Improvement in the **cost/performance ratio** of computer components
  - Hardware became more flexible
  - Introduction of **multiprocessing**, which allowed executing programs in parallel
  - Evolution of personal computers and high-speed communications
  - Introduction of **distributed processing** and **networked systems**

Brief History of Operating Systems Development (continued)

- **1990s:**
  - Demand for Internet capability sparked the proliferation of networking capability
  - Increased networking also created increased demand for tighter security 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)

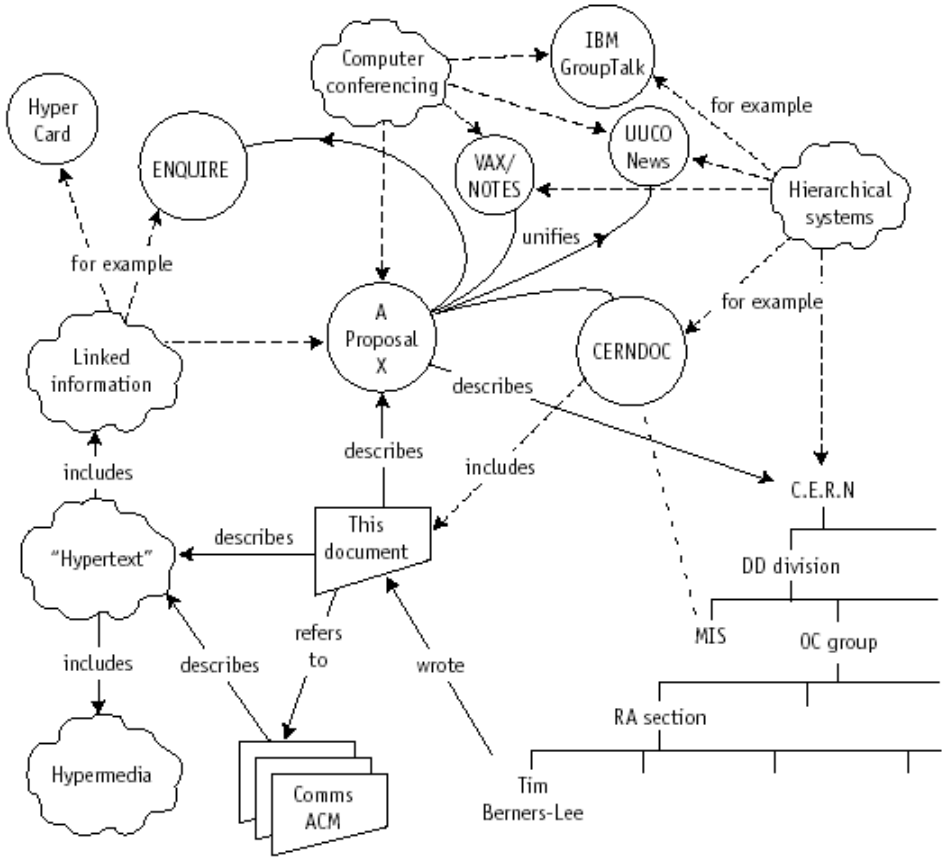


Figure 1.9: Linked information system by Tim Berners-Lee

Redux.epita.cf

Current Operating Systems

- Primary design features of current operating systems are based on providing support for
  - Multimedia applications
  - Internet and Web access
  - Client/server computing
- Computer systems are required to have
  - Increased CPU speed
  - High-speed network attachments
  - Increased number and variety of storage devices

System Architecture

- Improvements in system architecture
  - Use of object-oriented design
    - Possible to modify and customize pieces of an operating system without disrupting the integrity of the remainder of the system
    - Makes software development groups more productive
  - Reorganisation of the operating system's kernel
    - Limited to a few essential functions
      - process scheduling
      - memory allocation

System Architecture (continued)

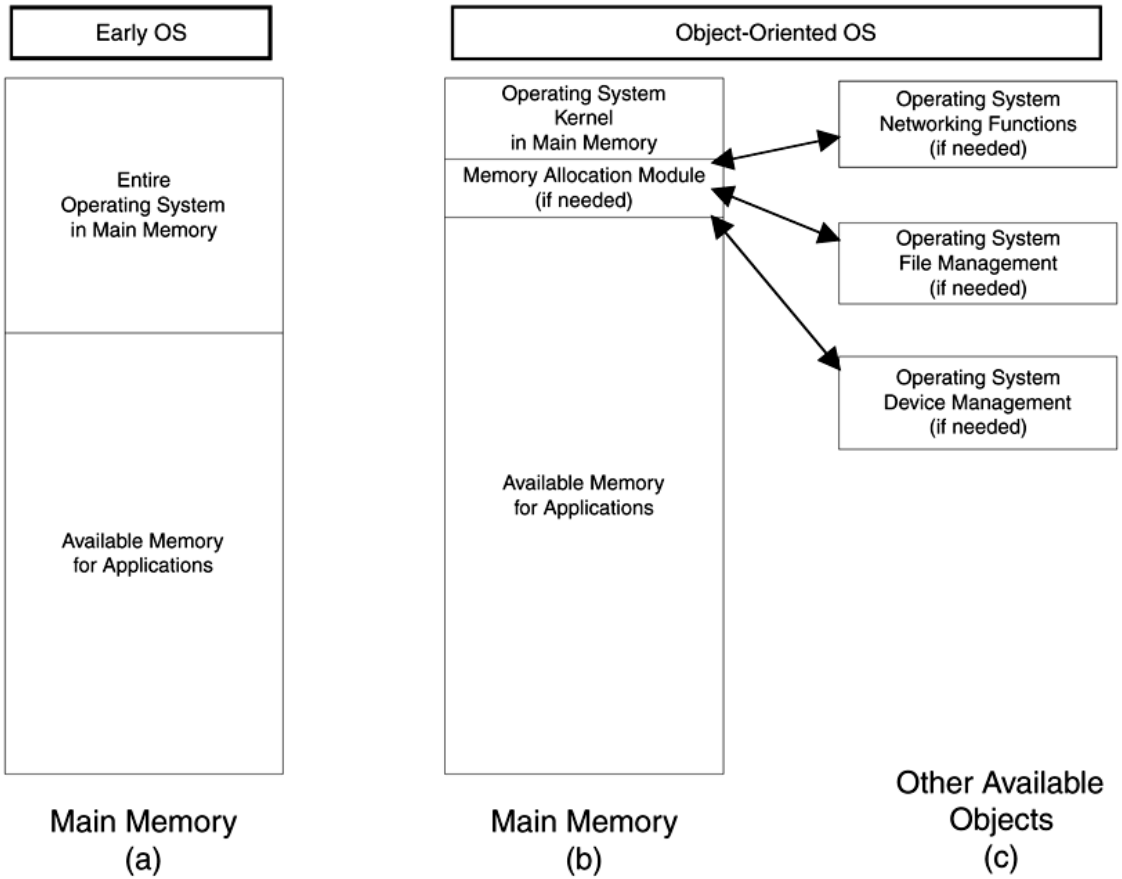


Figure 1.10: (a) Early operating systems; (b) & (c) Modern object-oriented systems

Redux.epita.cf

Threads

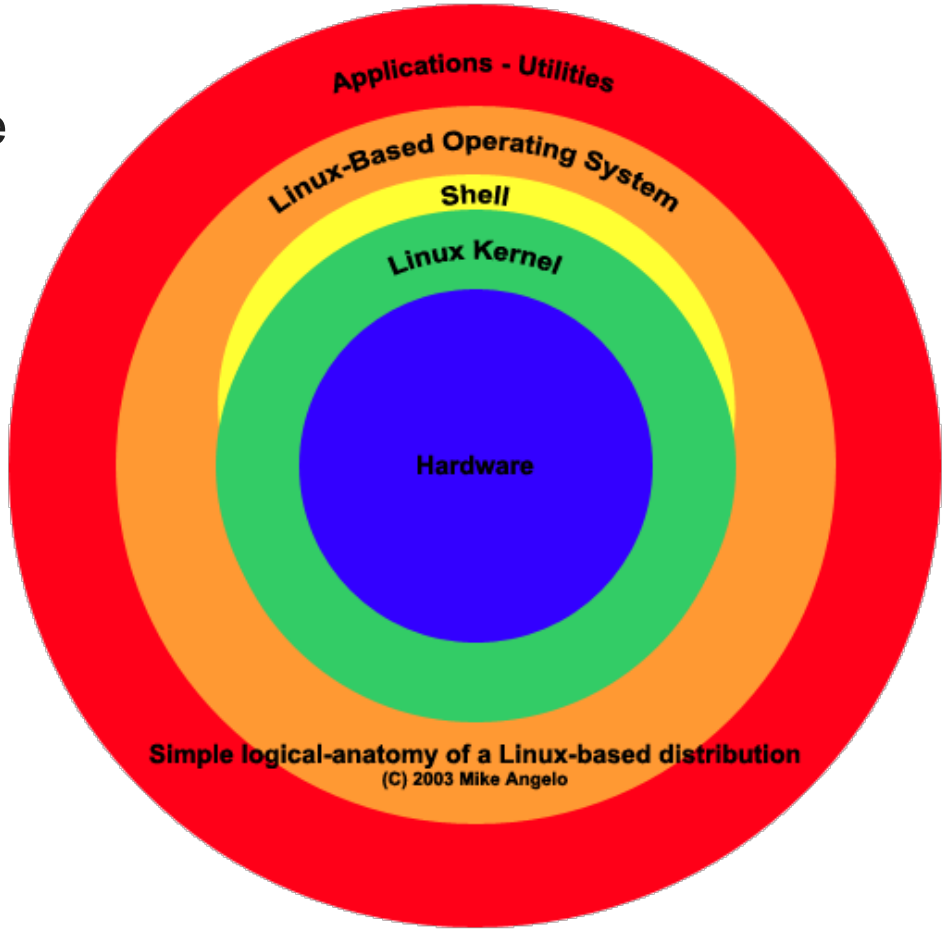
- **Thread:** A portion of a program that can run independently of other portions
  - The heavyweight process which owns the resources becomes a more passive element
  - Thread becomes the element that uses the CPU and is scheduled for execution
  - Swapping threads is less time consuming than swapping processes
- Multithreaded applications programs can have several threads running at one time with the same or different priorities

Multiprocessing Configurations

- **Symmetric multiprocessing:**
  - Allows for several CPUs to process multiple jobs at the same time
  - CPUs are independent of one another, but each has access to the operating system
- **Asymmetric multiprocessing:**
  - Some operating systems functions are assigned to subordinate processors, which take their instructions from the main CPU

Kernel

- **Kernel** – The internal part of the operating system.
  - Those software components that perform the basic functions required by the computer.
    - File management
    - Memory management (RAM)
    - Security



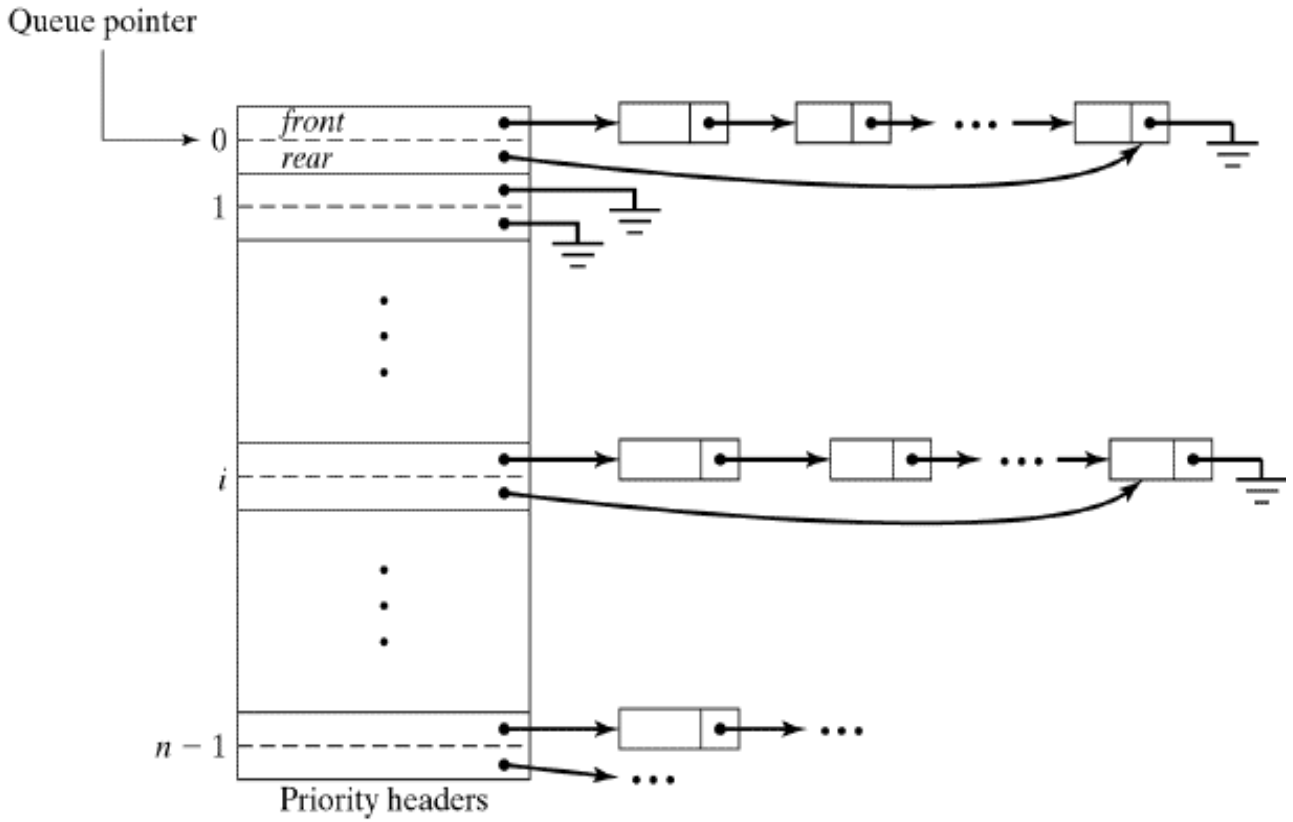
Kernel Definitions and Objects

- Basic set of objects, primitives, data structures, processes
- Rest of OS is built on top of kernel
- Kernel defines/provides ***mechanisms*** to implement various ***policies***
  - Process and thread management
  - Interrupt and trap handling
  - Resource management
  - Input/output

Queues

- OS needs many different queues
- Single-level queues
  - Implemented as array
    - Fixed size
    - Efficient for simple FIFO operations
  - Implemented as linked list
    - Unbounded size
    - More overhead, but more flexible operations

Priority Queues



Operating System Kernel

- “The one program running at all times on the computer” is the **kernel.** Everything else is either a system program (ships with the operating system) or an application program

Computer Startup

- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

Operating-System Operations

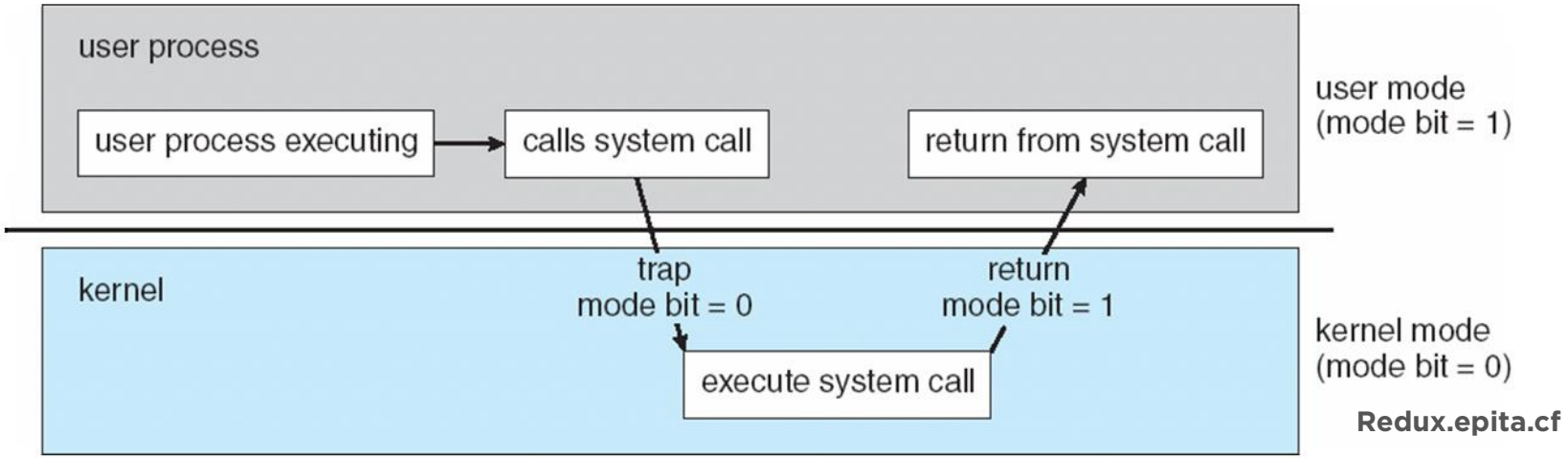
- Interrupt driven by hardware
- Software error or request creates **exception** or **trap**
  - Division by zero, request for operating system service
- Other process problems include infinite loop, processes modifying each other or the operating system

Operating-System Operations

- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **Mode bit** provided by hardware
    - Provides ability to distinguish when system is running user code or kernel code
    - Some instructions designated as **privileged**, only executable in kernel mode
    - System call changes mode to kernel, return from call resets it to user

Transition from User to Kernel Mode

- Timer to prevent infinite loop / process hogging resources
  - Set interrupt after specific period
  - Operating system decrements counter
  - When counter zero generate an interrupt
  - Set up before scheduling process to regain control or terminate program that exceeds allotted time



Multiprocessing Configurations (continued)

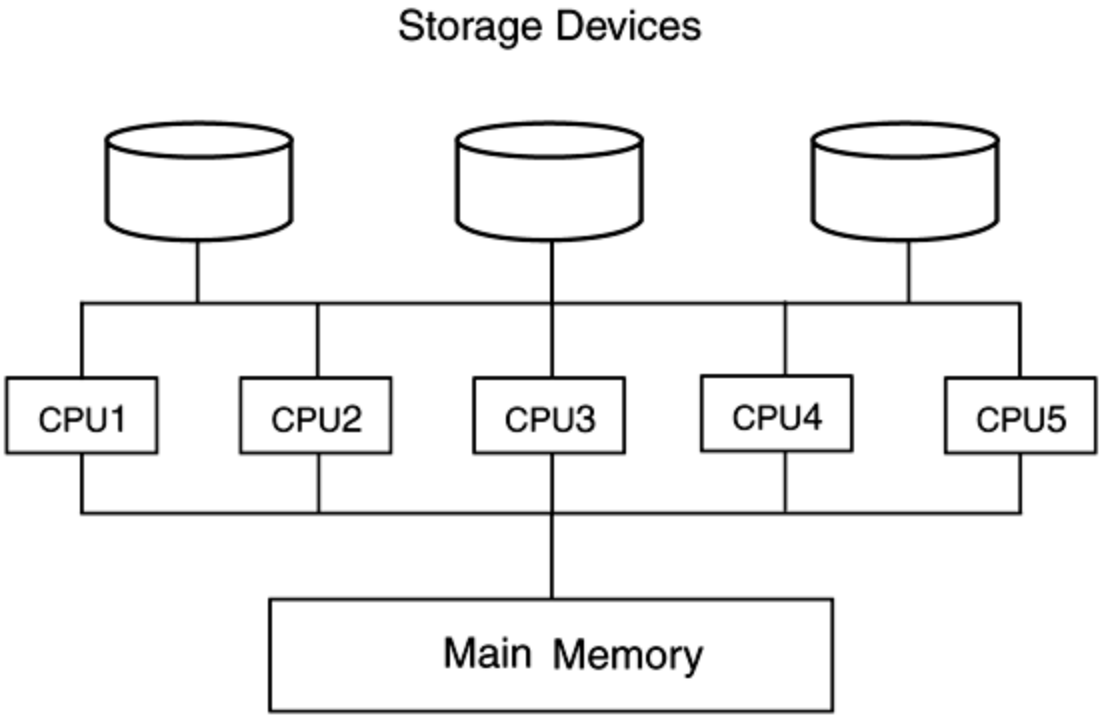


Figure 1.11: Symmetric multiprocessing system with five processors

Multiprocessing Configurations (continued)

- Network PCs gave impetus to the concept of **distributed processing**
  - Processors are placed at remote locations and are connected to each other via telecom devices
  - Different from symmetric multiprocessing systems as they do not share memory
  - Computations can be dispersed among several processors
  - Overall capability of the computer system is maximized

Summary

- Operating System manages all of the hardware and software of a computer system
- Each manager of an OS both works closely with the other managers and performs its unique role
- Operating systems with networking capability have Network Manager
- Essential hardware components include memory chips, I/O, storage devices and CPU
- Until mid-1970s, computers were classified by capacity and price

Summary (continued)

- Computing power has been rising exponentially—Moore's Law
- Dramatic changes in physical size, cost, and memory capacity with time
- Networking has become an integral part of modern computer systems
- Delivering information to a mobile society, creating a strong market for handheld devices
- Operating systems fall into following five categories: batch, interactive, real-time, hybrid and embedded

Summary (continued)

- Use of object-oriented design improved the system architecture
- Symmetric multiprocessing allows for several CPUs to process multiple jobs at the same time
- Network PCs gave impetus to the concept of distributed processing