Operating Systems

Chapter 3: Operating Systems

- 3.1 The History of Operating Systems
- 3.2 Operating System Architecture
- 3.3 Coordinating the Machine's Activities
- 3.4 The concept of a Process

Types of Software

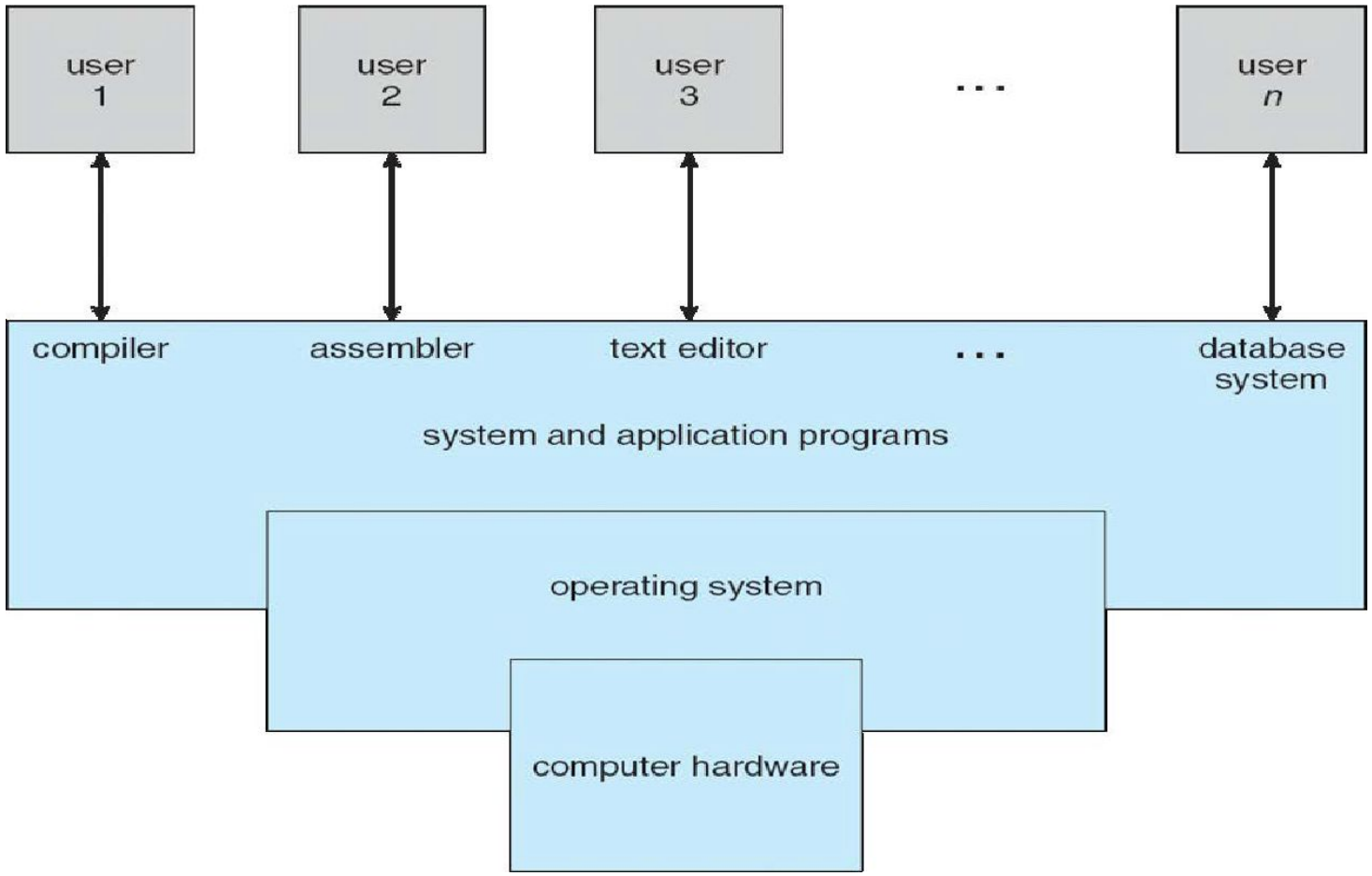
- Application software
  - Performs specific tasks for users
  - e.g., MS word, chrome
- System software
  - Provides infrastructure for application software
  - Program designed to run a computer's hardware and applications and manage its resources, such as its memory, processors, and devices.
  - Consists of operating system and utility software

Operating Systems

- A program that acts as an intermediary between a user of a computer and the computer hardware.

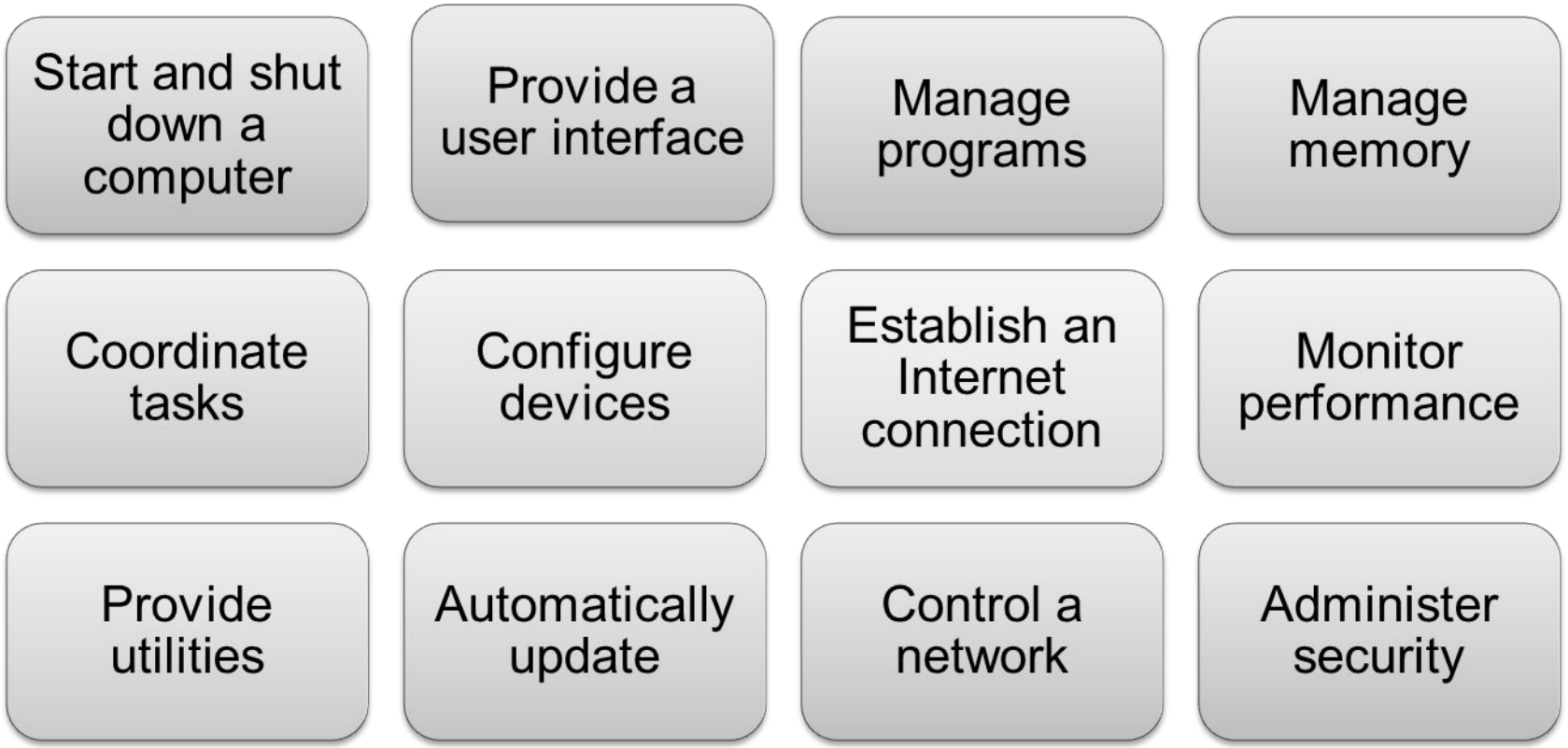
- Operating system goals:
  - Execute user programs and make solving user problems easier.
  - Make the computer system convenient to use.
  - Use the computer hardware in an efficient manner.

Four Components of a Computer System



Different Operating Systems

Functions of Operating Systems



Operating System Structure

- Batch processing
  - equires multiprogramming
- Interactive processing
  - equires real-time processing
- Time-sharing/Multitasking
  - Implemented by Multiprogramming
- Multiprocessor machines

Figure 3.1 **Batch processing**

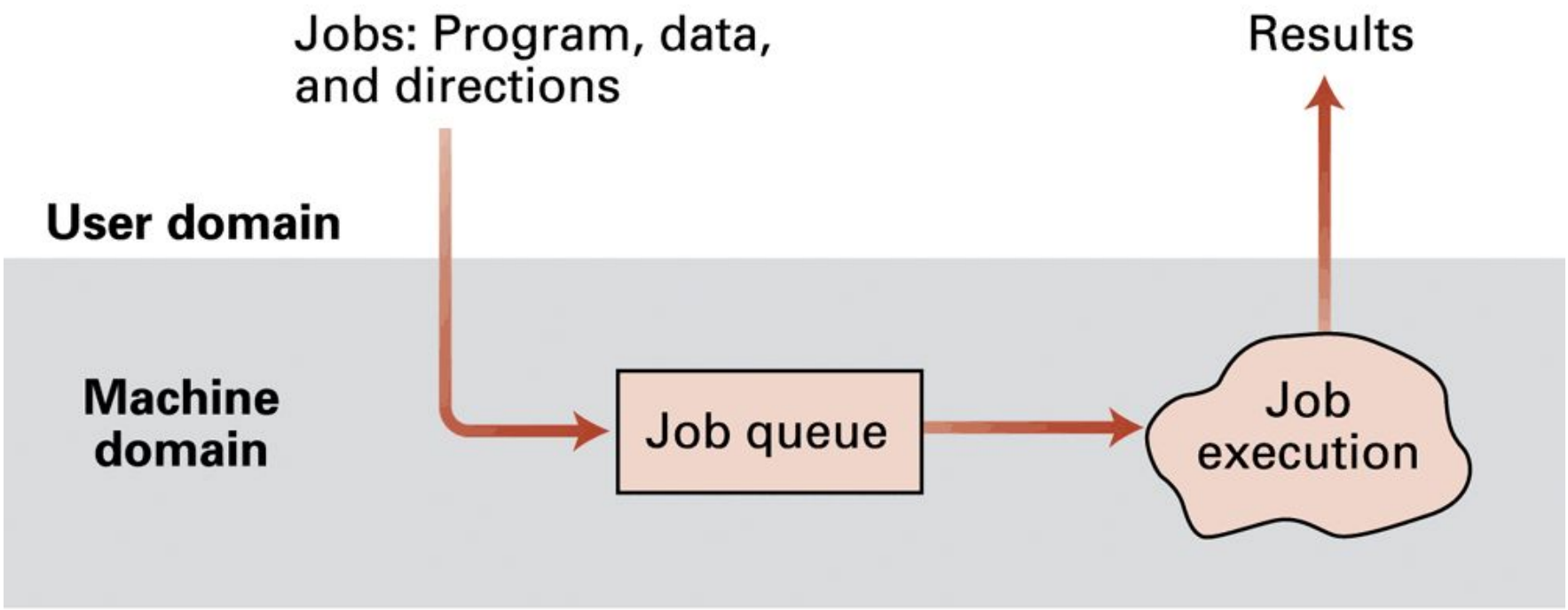


Figure 3.2 Interactive processing

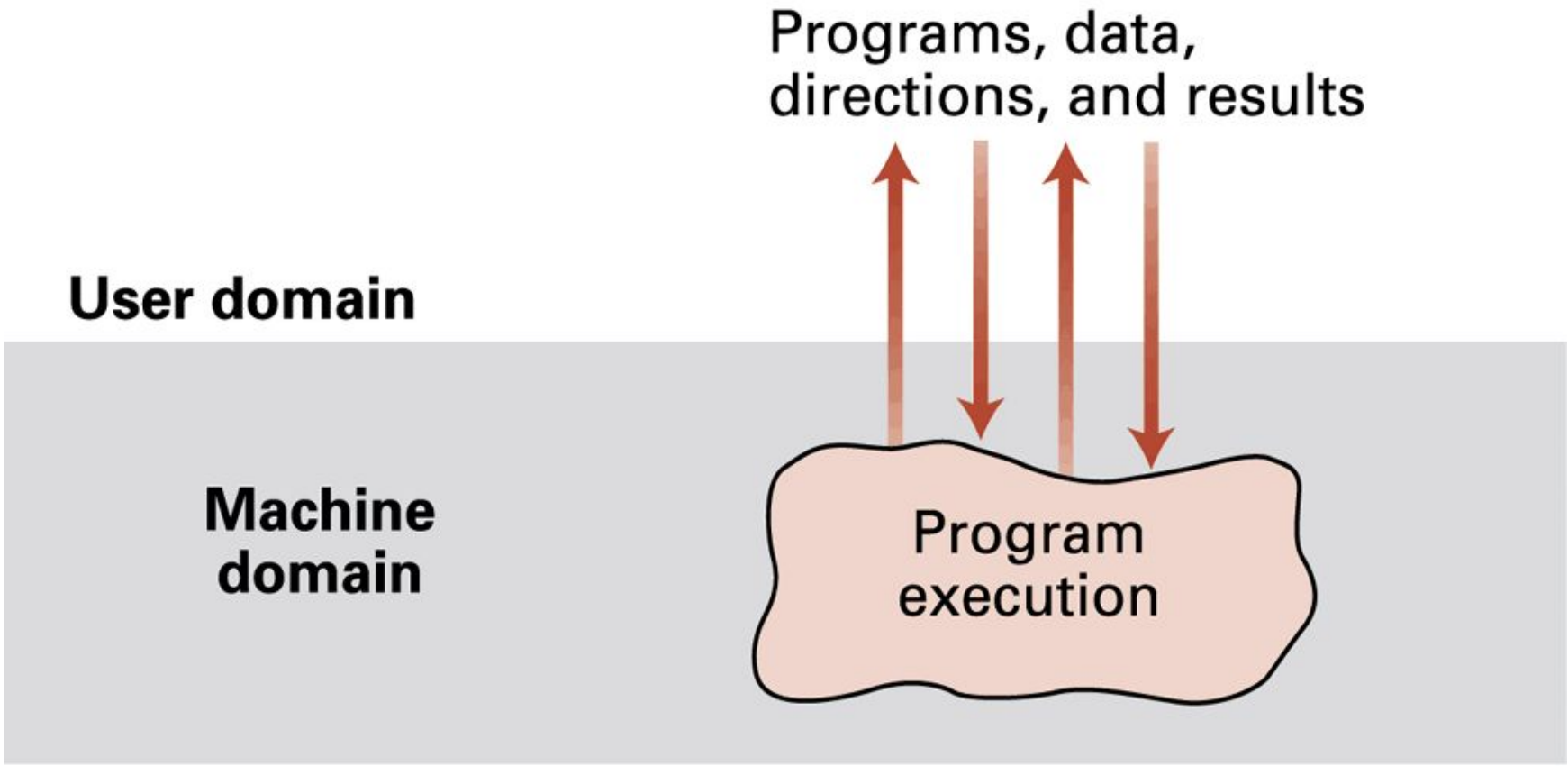
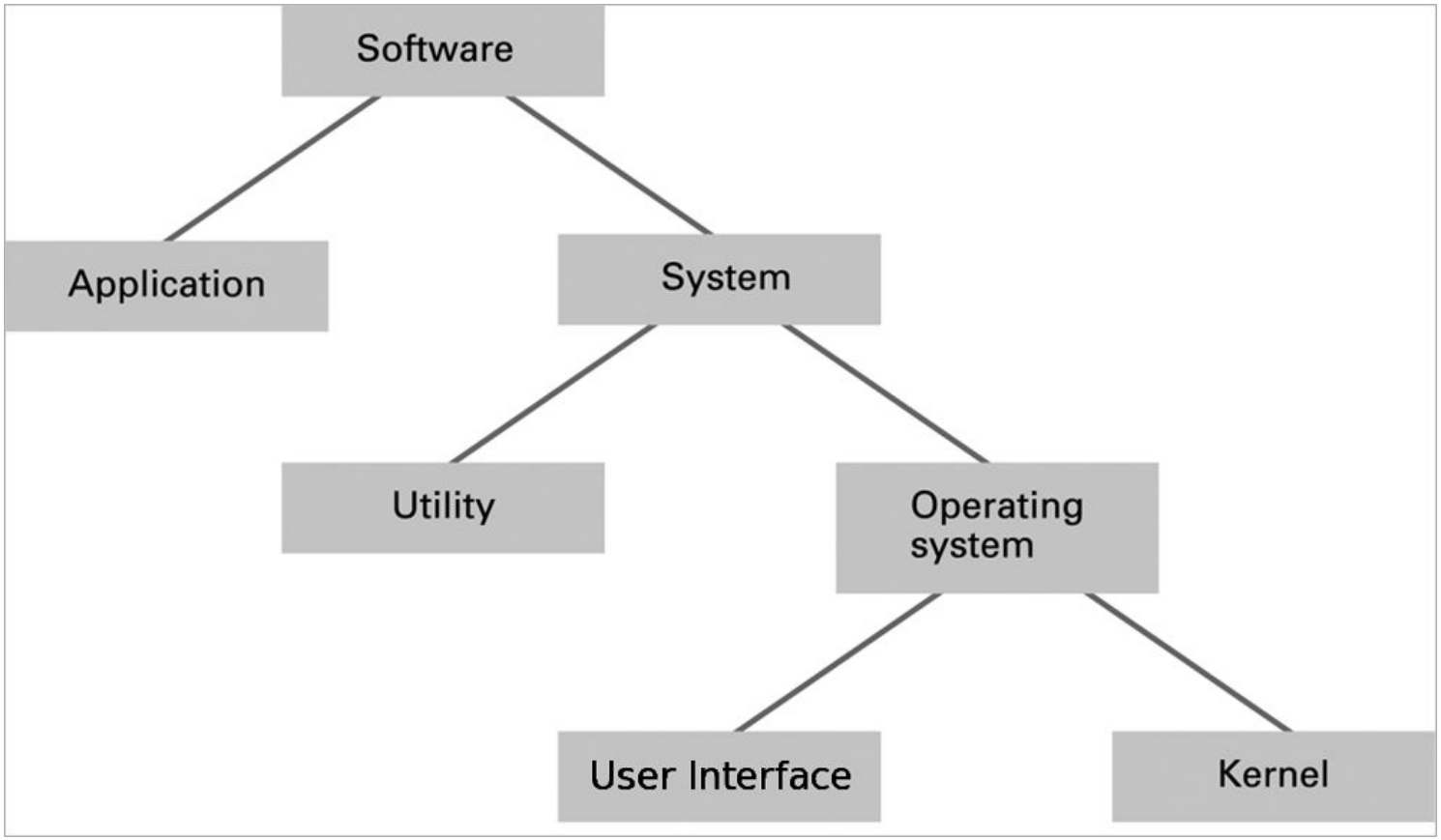


Figure 3.3 **Software classification**



Operating System Functions

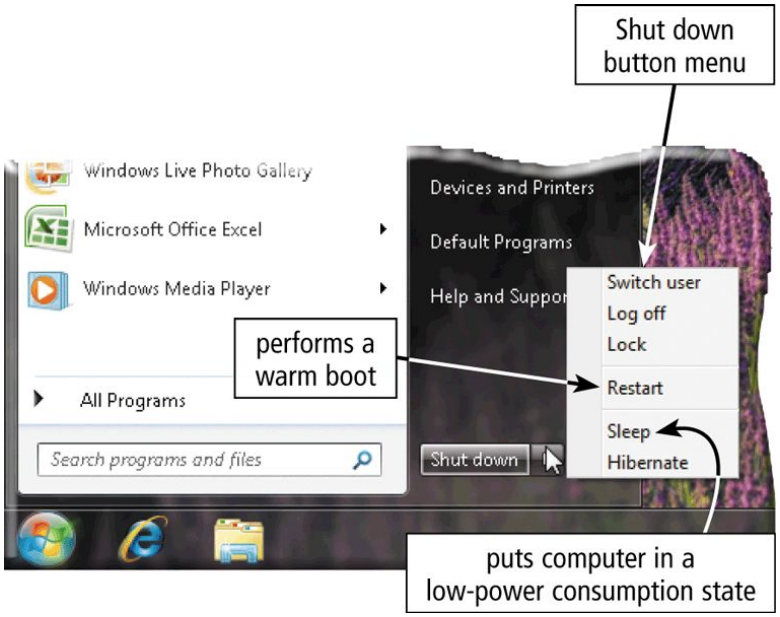
- The initial program that runs when a computer is powered up is called the **bootstrap program**, stored in ROM.
- The process of starting or restarting a computer is called **booting.**

Cold boot

- Turning on a computer that has been powered off completely

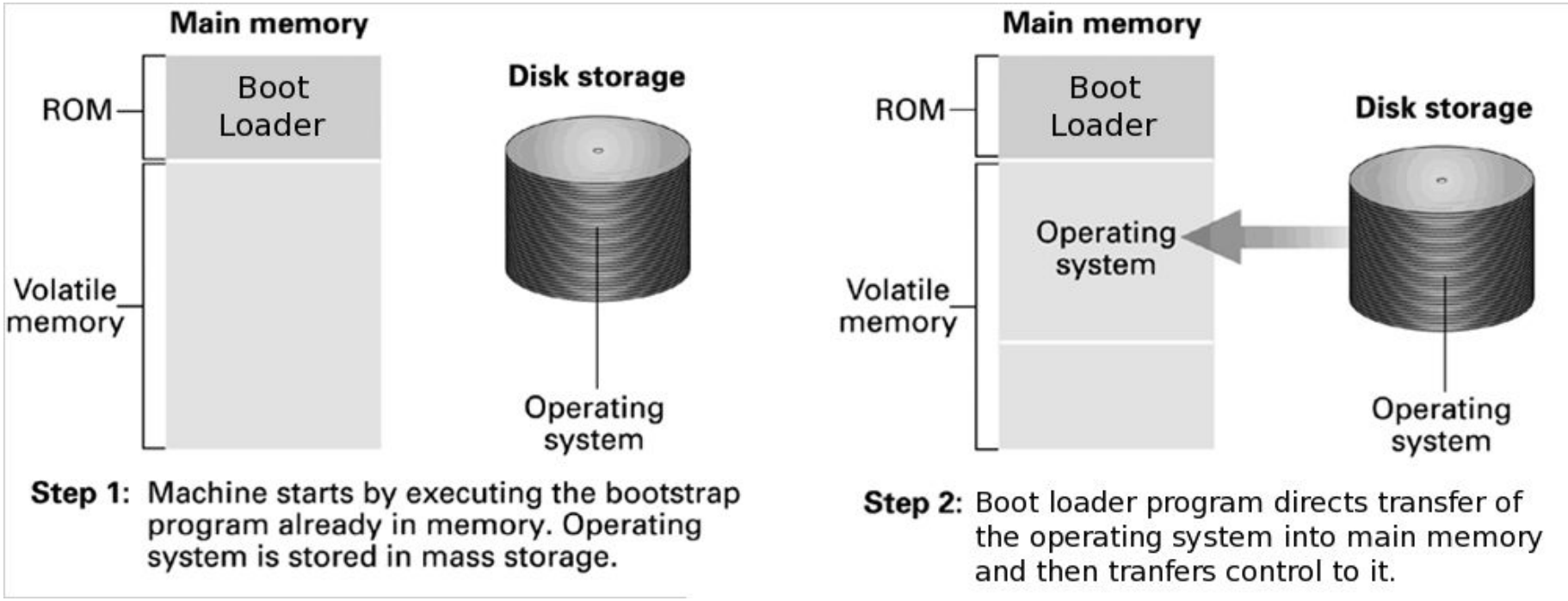
Warm boot

- Using the operating system to restart a computer



Page 400
Figure 8-2

Figure 3.4 The booting process



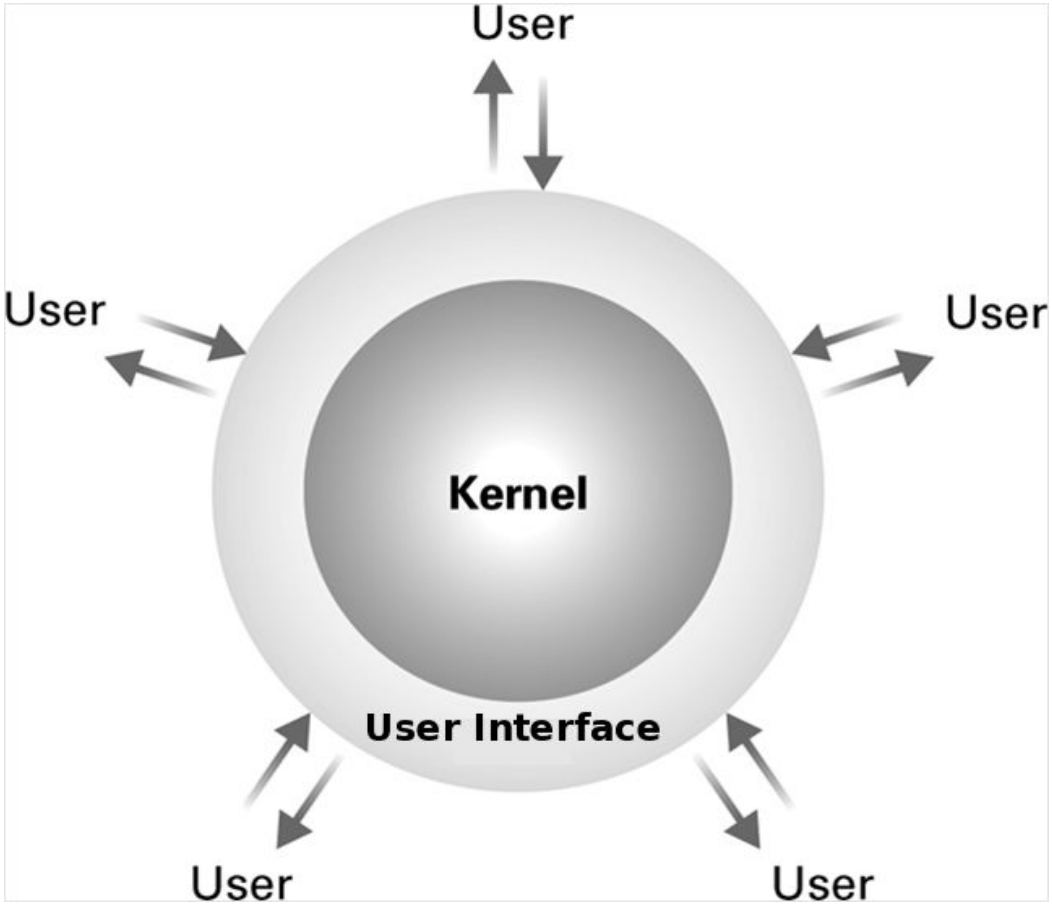
Getting it Started (Bootstrapping)

- **Boot loader:** Program in ROM
  - Run by the CPU when power is turned on
  - Transfers operating system from mass storage to main memory
  - Executes jump to operating system

Operating System Components

- **User Interface:** Communicates with users
  - Text based (Shell)
  - Graphical user interface (GUI)
- **Kernel:** The one program running at all times on the computer. Performs basic required functions
  - File manager
  - Device drivers
  - Memory manager
  - Scheduler and dispatcher

Figure 3.5 The user interface act as an intermediary between users and the operating system kernel



File Manager

- **Directory** (or **Folder**): A user-created bundle of files and other directories (subdirectories)
- **Directory Path:** A sequence of directories within directories

Memory Manager

- Allocates space in main memory
- May create the illusion that the machine has more memory than it actually does (**virtual memory**) by playing a "shell game" in which blocks of data (**pages**) are shifted back and forth between main memory and mass storage

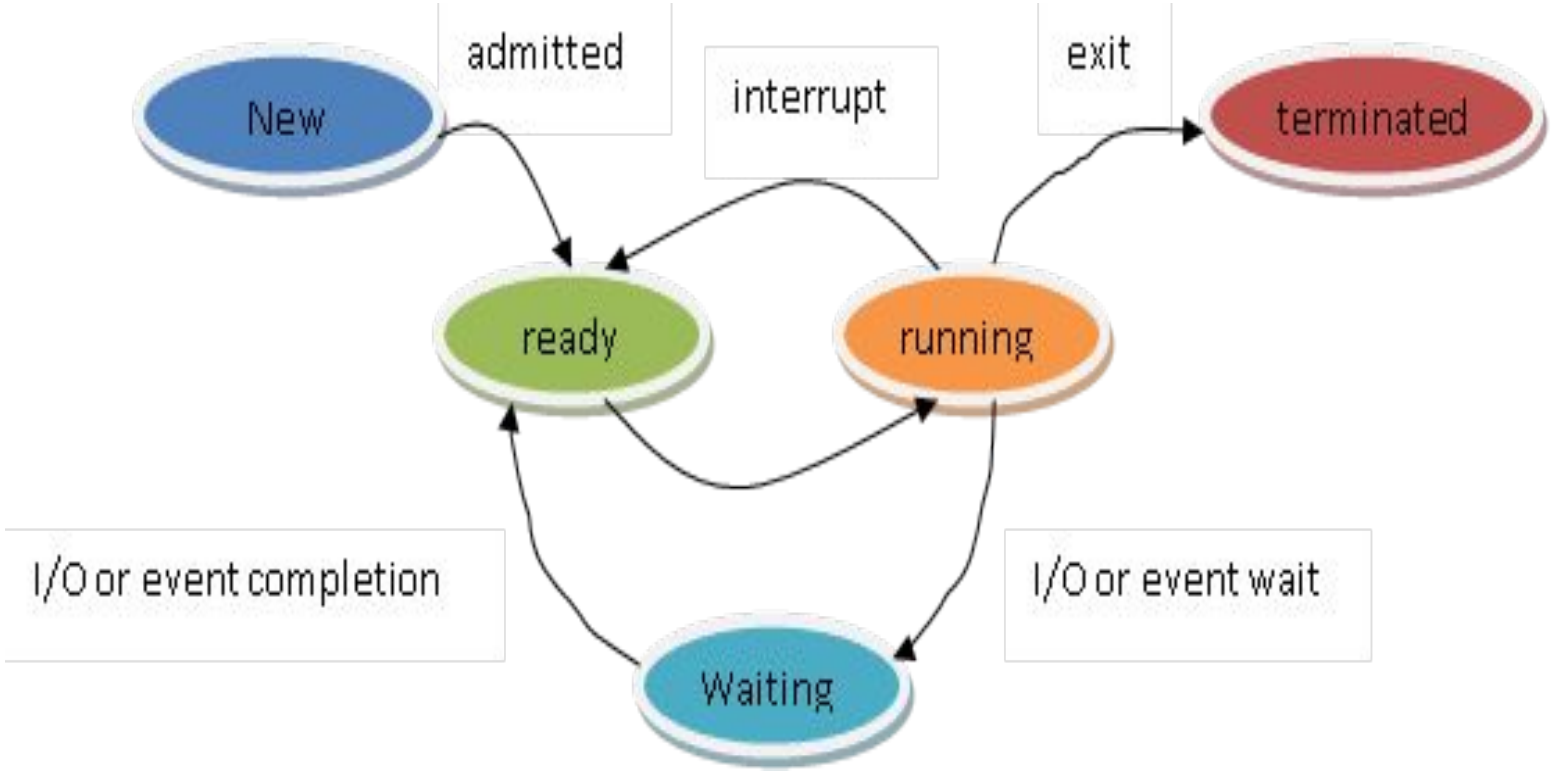
Processes

- **Process:** The activity of executing a program
- **Process Table:** To keep track of all the processes, the scheduler maintains a block of information in main memory.
  - Program counter
  - General purpose registers
  - Related portion of main memory

Process State

- A process changes its state during its execution. Each process may be in one of the following states:
- **New:** when a new process is being created.
- **Running:** A process is said to be in running state when instructions are being executed.
- **Waiting:** The process is waiting for some event to occur (such as an I/O operation).
- **Ready:** The process is waiting for processor.
- **Terminated:** The process has finished execution.

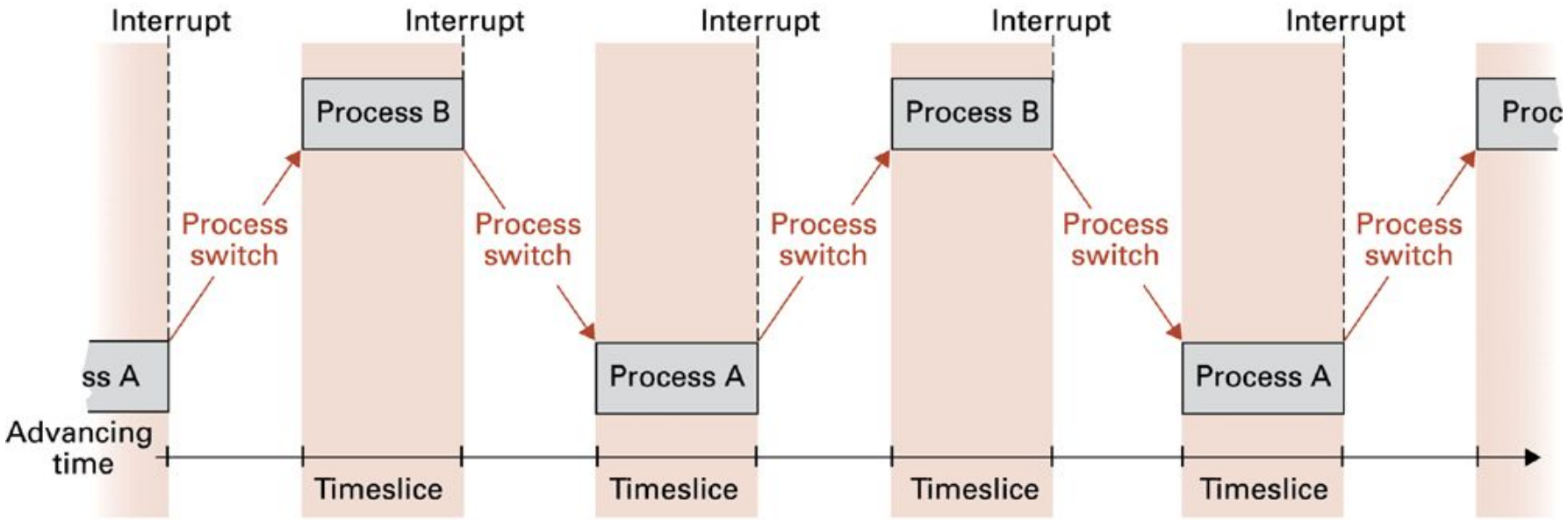
Process State Diagram



Process Administration

- **Scheduler:** Adds new processes to the process table and removes completed processes from the process table.
  - Responsible for determining which process should run next on the CPU.
- **Dispatcher:** Controls the allocation of time slices to the processes in the process table
  - The end of a time slice is signaled by an interrupt.

Figure 3.6 Time-sharing between process A and process B



Inter-Process Communication (IPC) (Client-Server Model):

- A mechanism that allows processes to communicate and synchronize their actions. It can be implemented using various methods, such as message passing, shared memory, and semaphores.
- **Client-Server Model:**
- **Client:** A process that requests services or resources from another process (the server).
- **Server:** A process that provides services or resources to clients. Servers typically handle multiple requests from multiple clients, often through sockets or other network protocols.