

# **Chapter 3:**

# **Operating Systems**

**Computer Science: An Overview**  
**Twelfth Edition**

**by**  
**J. Glenn Brookshear**  
**Dennis Brylow**

# 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 Handling Competition Among Processes\*
- 3.5 Security

\* Not covered

# Functions of Operating Systems

An **operating system** is the software that controls the overall operation of a computer.

- Oversee operation of computer
- Store and retrieve files
- Schedule programs for execution
- Coordinate the execution of programs

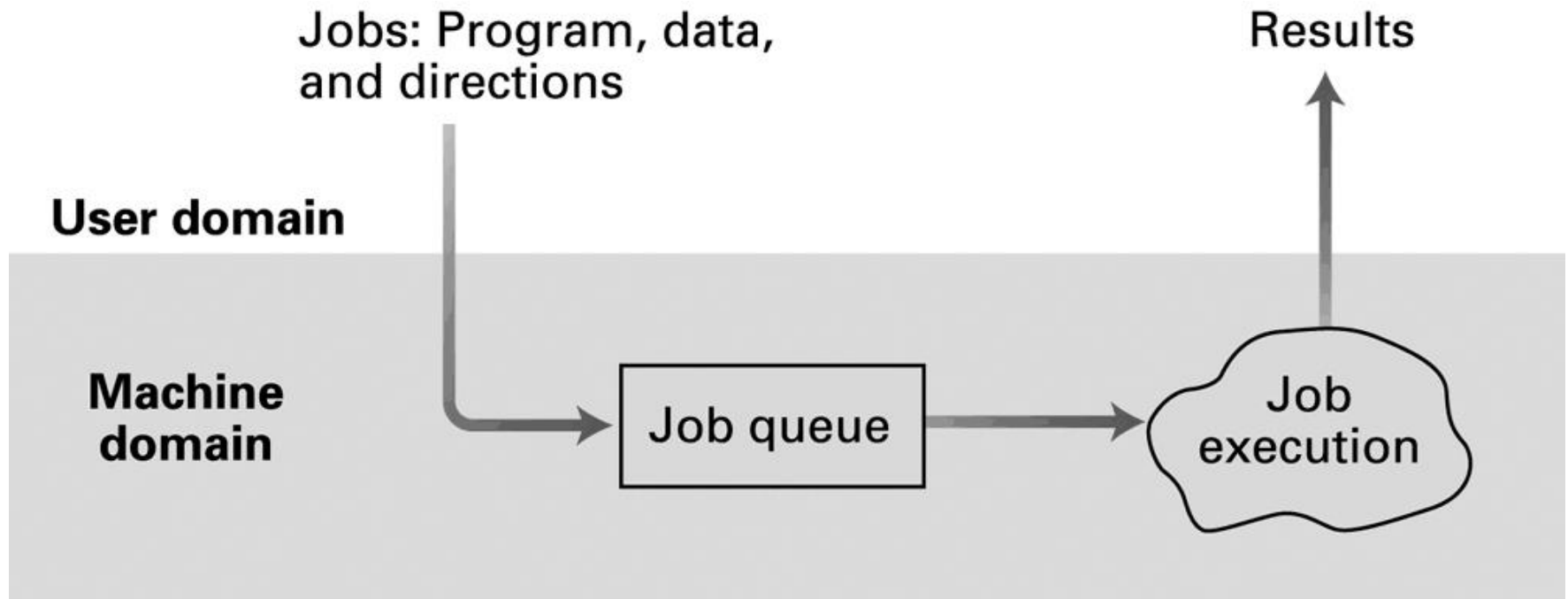
# The History of Operating Systems

- In 1940s and 1950s, computers were not very flexible or efficient
- Program execution consisted of mounting magnetic tapes, placing punched cards in card readers, setting switches
- Sign-up sheets for multiple users
- Computer operators were hired to eliminate physical contact with the users

# Evolution of Shared Computing

- **Batch processing** - the execution of jobs by collecting them in a single batch, then executing them without further interaction with the user.
- Job queue, first-in first-out
- Job control language (JCL)
- Major drawback of operator based batch processing: users have no interaction with their jobs

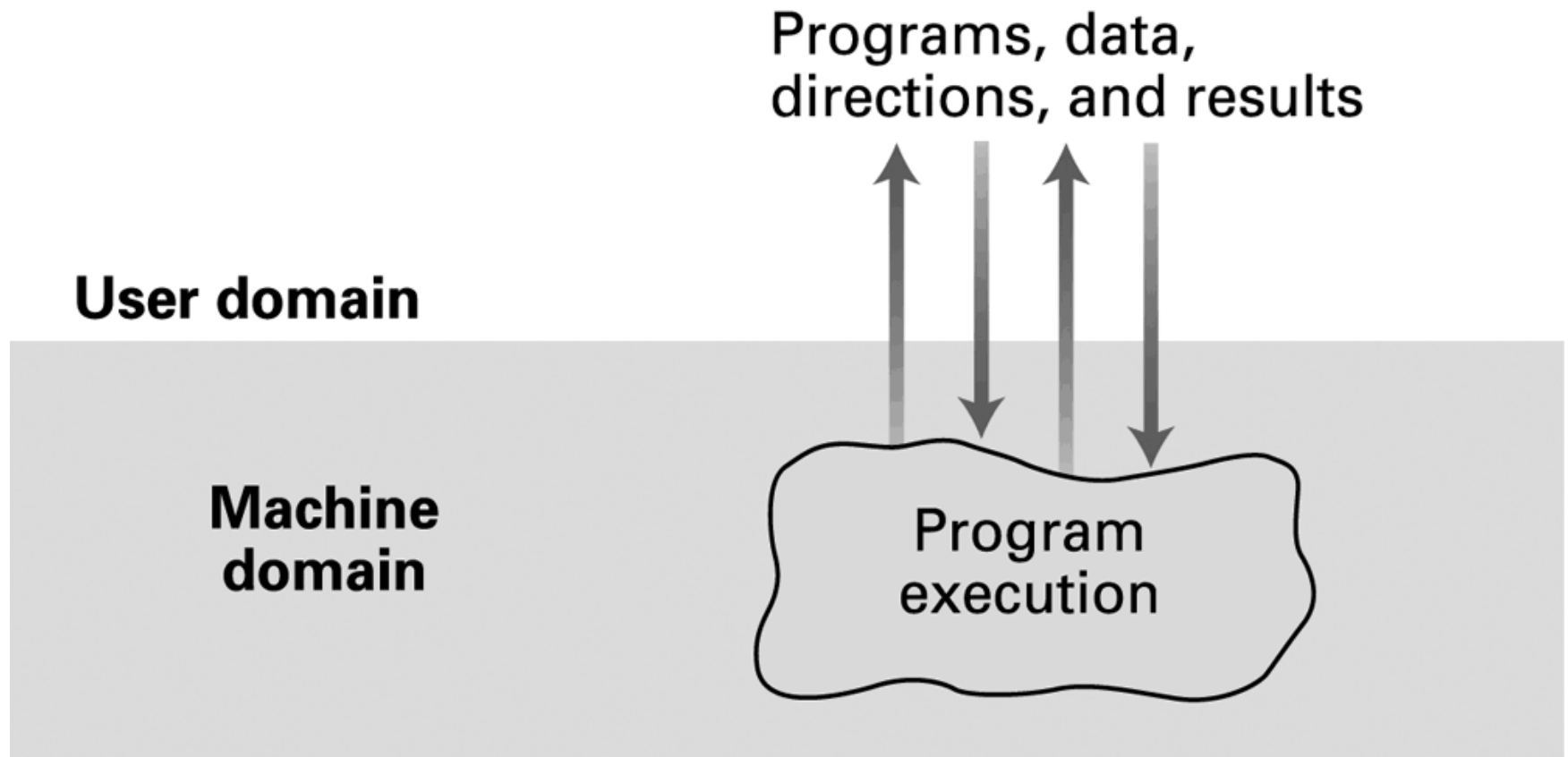
# Figure 3.1 Batch processing



# Evolution of Shared Computing

- Interactive processing
  - Requires real-time processing
  - the computer performs the task in accordance with deadlines in its environment
  - Problem: one user at a time in multi-user system
- Time-sharing/Multitasking
  - Implemented by Multiprogramming
  - time-sharing refers to multiple users sharing access to a common computer,
  - multitasking refers to one user executing numerous tasks simultaneously.

## Figure 3.2 Interactive processing





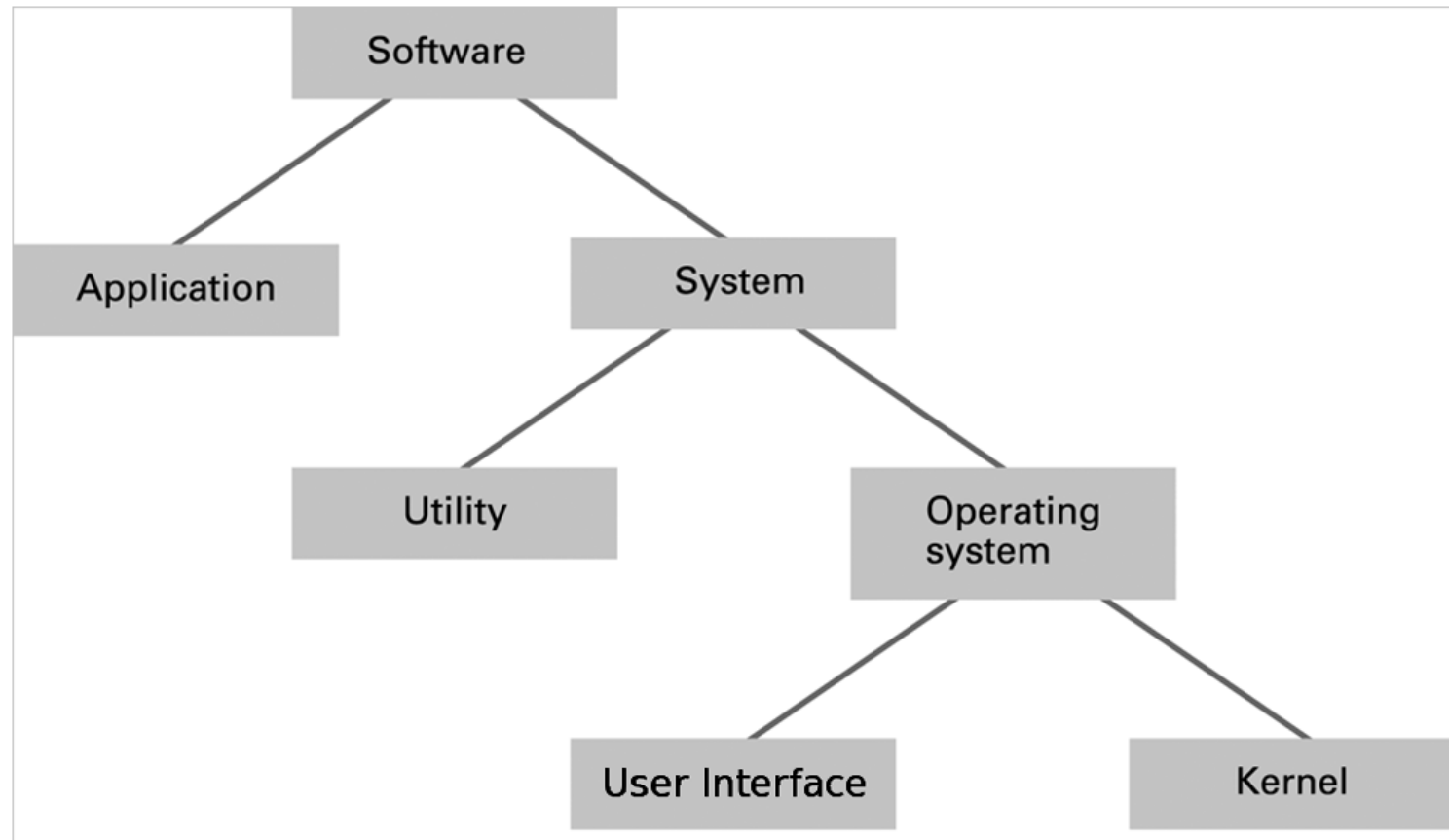
# Evolution of Shared Computing

- Multiprocessor machines
- Problems
  - **Load balancing:** dynamically allocating tasks to the various processors so that all processors are used efficiently
  - **Scaling:** breaking tasks into a number of subtasks compatible with the number of processors available

# Types of Software

- Application software
  - Performs specific tasks for users
- System software
  - Provides infrastructure for application software
  - Consists of operating system and utility software. Examples: disk formatter, cd copier, compress and decompress data, software for playing multimedia

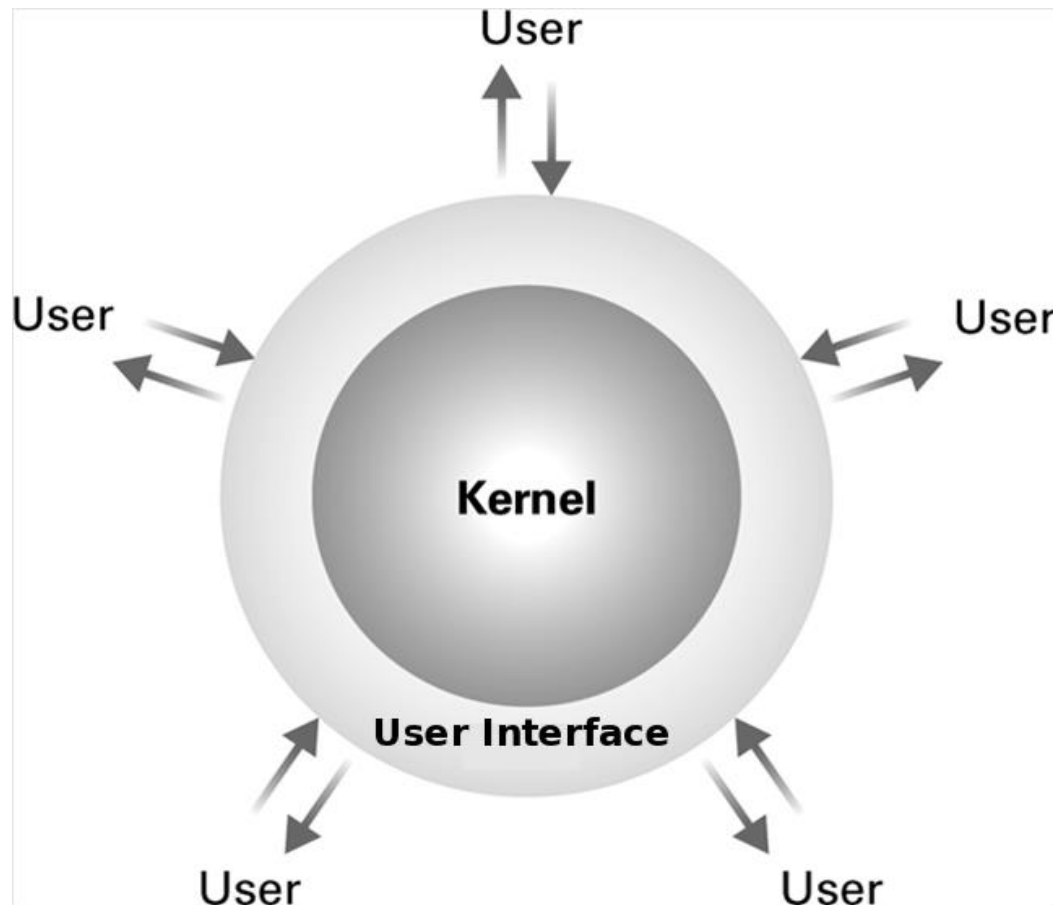
# Figure 3.3 Software classification



# Operating System Components

- **User Interface:** Communicates with users
  - Text based (Shell)
  - Graphical user interface (GUI)
- **Kernel:** Performs basic required functions
  - File manager
  - Device drivers
  - Memory manager
  - Scheduler and dispatcher

# Figure 3.4 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

# Device Drivers

- the software units that communicate with the controllers
- uniquely designed for its particular type of device
- users do not need the hardware's details

# Memory Manager

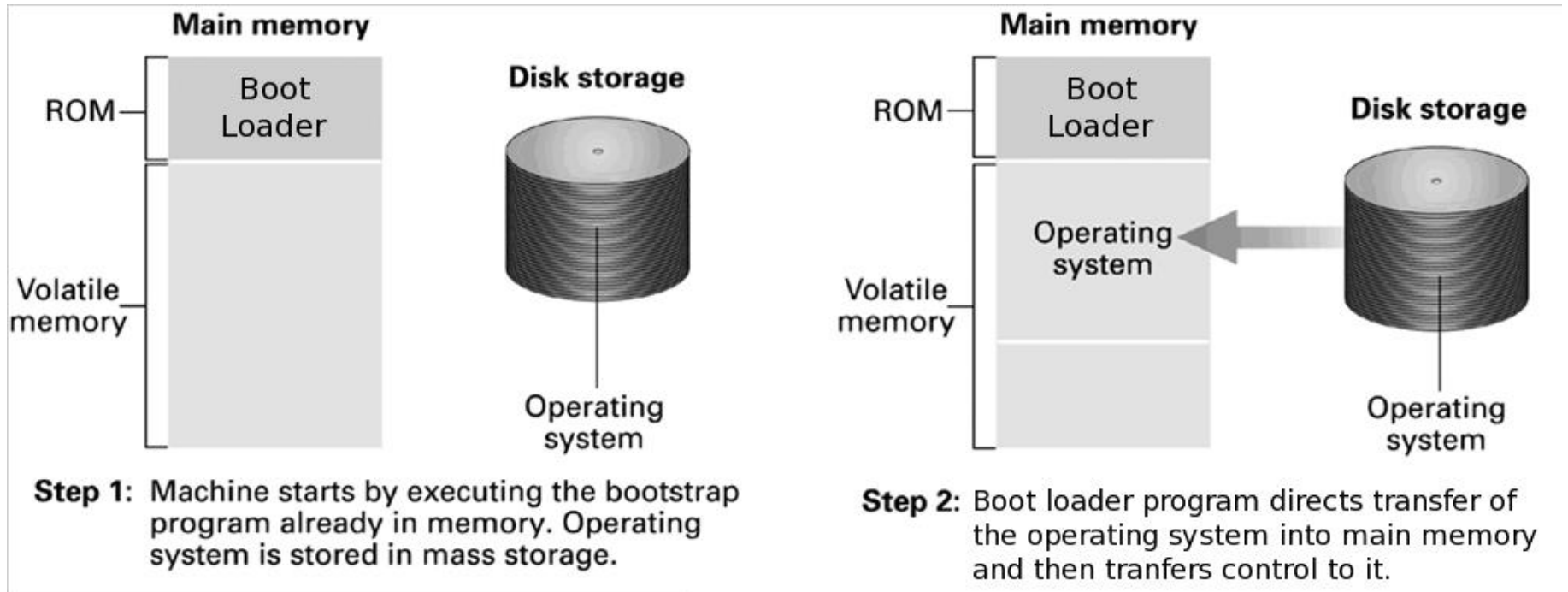
- Allocates space in main memory
- May create the illusion that the machine has more memory than it actually does (**virtual memory**) by playing a “shell game” in which blocks of data (**pages**) are shifted back and forth between main memory and mass storage



# Getting it Started (Bootstrapping)

- **Boot loader:** Program in ROM (example of firmware) (to find the initial program counter)
  - Run by the CPU when power is turned on
  - Transfers operating system from mass storage to main memory
  - Executes jump to operating system

# Figure 3.5 The booting process



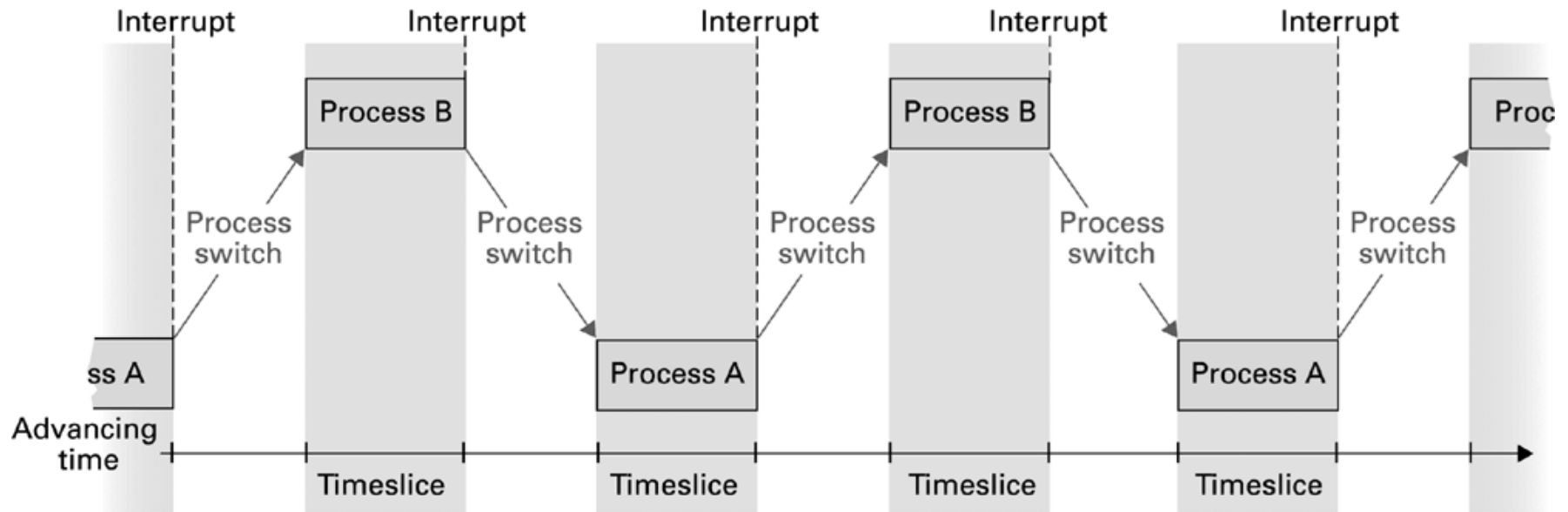
# Processes

- **Process:** The activity of executing a program
- **Process State:** Current status of the activity
  - Program counter
  - General purpose registers
  - Related portion of main memory

# Process Administration

- **Scheduler:** Adds new processes to the process table and removes completed processes from the process table
  - Each process has assigned memory, priority and status (ready/waiting) information.
- **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



# Security

- Attacks from outside
  - Problems
    - Insecure passwords
    - Sniffing software
  - Counter measures
    - Auditing software

# Security (continued)

- Attacks from within
  - Problem: Unruly processes
  - Counter measures: Control process activities via privileged modes and privileged instructions
  - privileged instructions change the contents of memory limit registers and the current privilege mode of the CPU