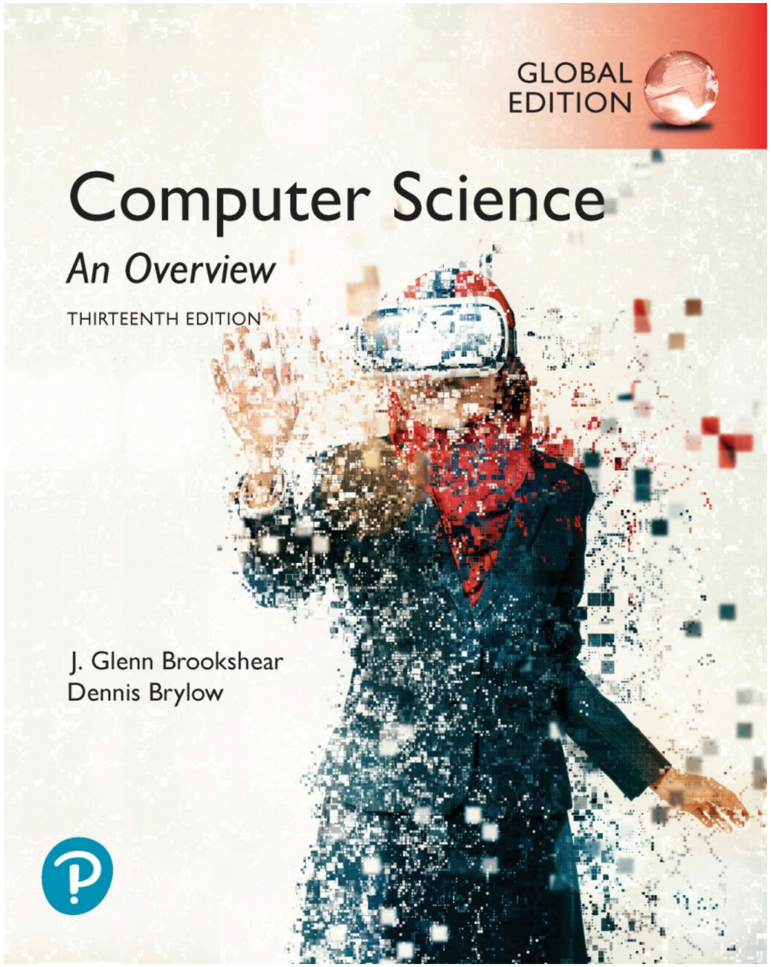
Computer Science An Overview

13th Edition, Global Edition



Chapter 3

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 Handling Competition Among Processes
- 3.5 Security

Pearson

Examples of Operating Systems

- Windows
- UNIX
- Mac OS
- Solaris (Sun/Oracle machines)
- Linux

Smartphone Operating Systems

- Apple iOS
- Windows Phone
- BlackBerry OS
- Nokia Symbian OS
- Google Android

Functions of Operating Systems

- Oversee operation of computer 監視电腦操作
- Store and retrieve files 儲存、存取 檔案
- Provide the user interface to request execution of programs 提供使用者界面 來執行程式
- Coordinate the execution of programs

3.1 History of Operating Systems

- Each program is called a “job”
- Early computers required significant setup time
- Each “job” required its own setup
- Operating Systems began as systems for simplifying setup and transitions between jobs

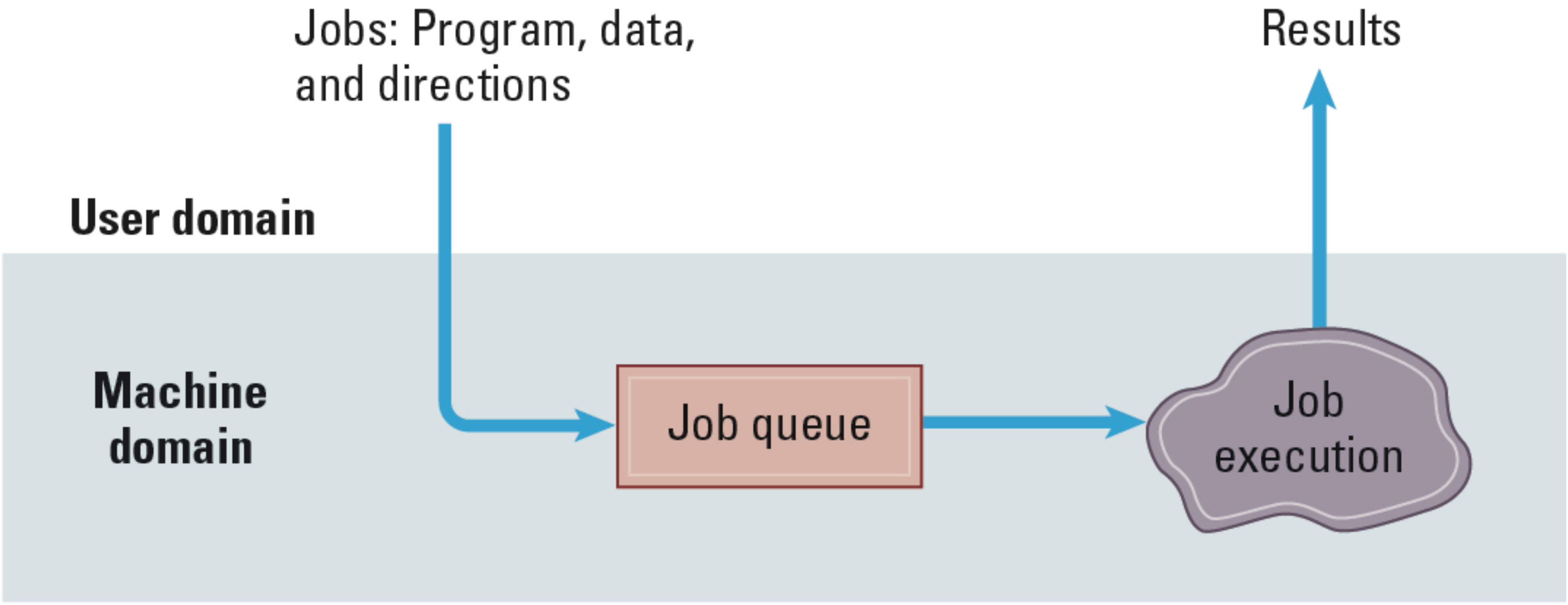
3.1 History of Operating Systems

Job queue P.165

- Batch processing (job queue) 批次處理任務 FIFO(先進先出)特性
- Interactive processing (real time) 交互式處理 (互動式處理)
- Time-sharing (one machine, many users) 分時系統 figure 3.6 ppt P.21
- Multitasking (one user, many tasks) 多工處理
- Multiprocessor machines (load balancing) 負載平衡 多元處理 (多處理機系統)
- Embedded Systems (specific devices) 嵌入式系統

Figure 3.1 Batch processing

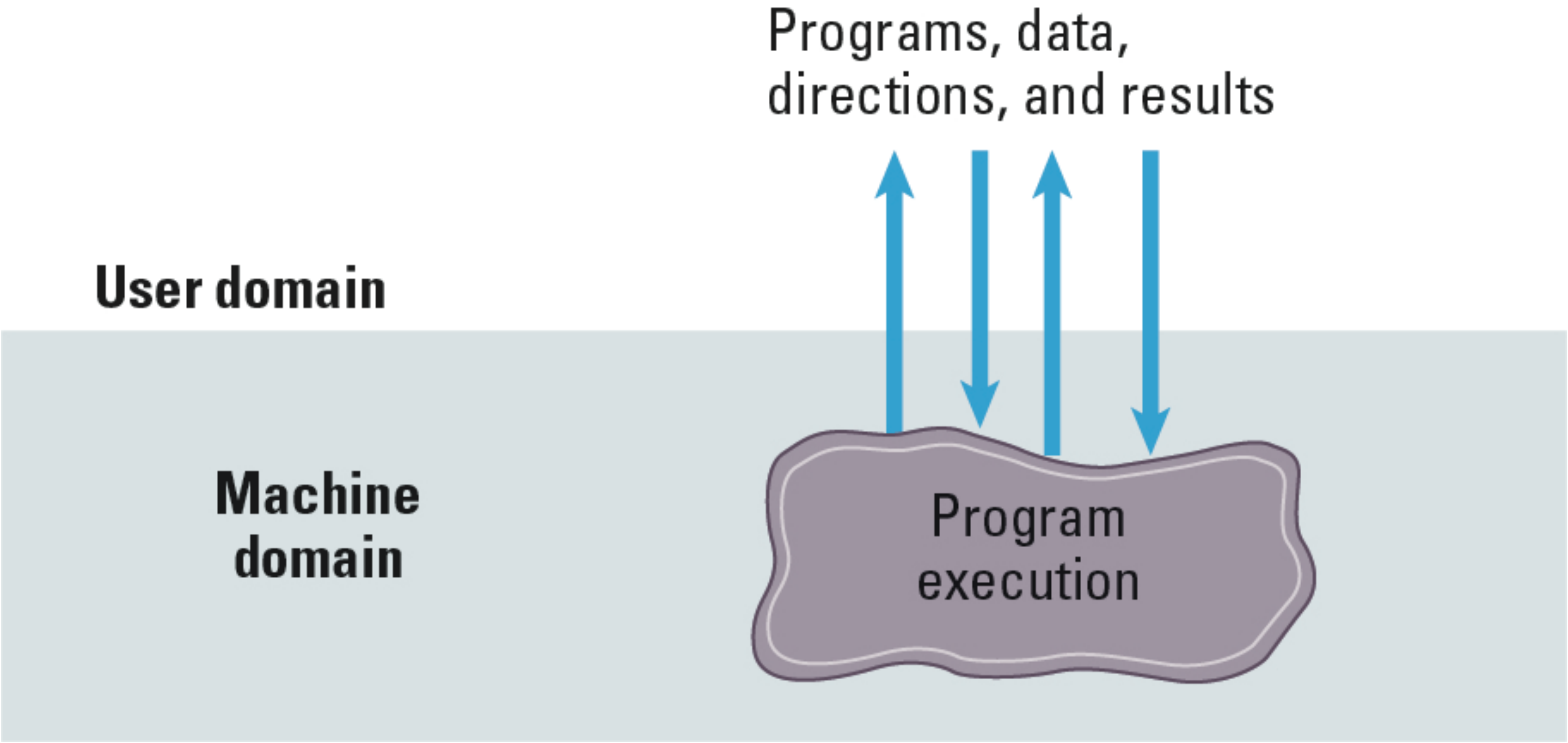
批次任務處理



交互式處理

(互动式處理)

Figure 3.2 Interactive processing

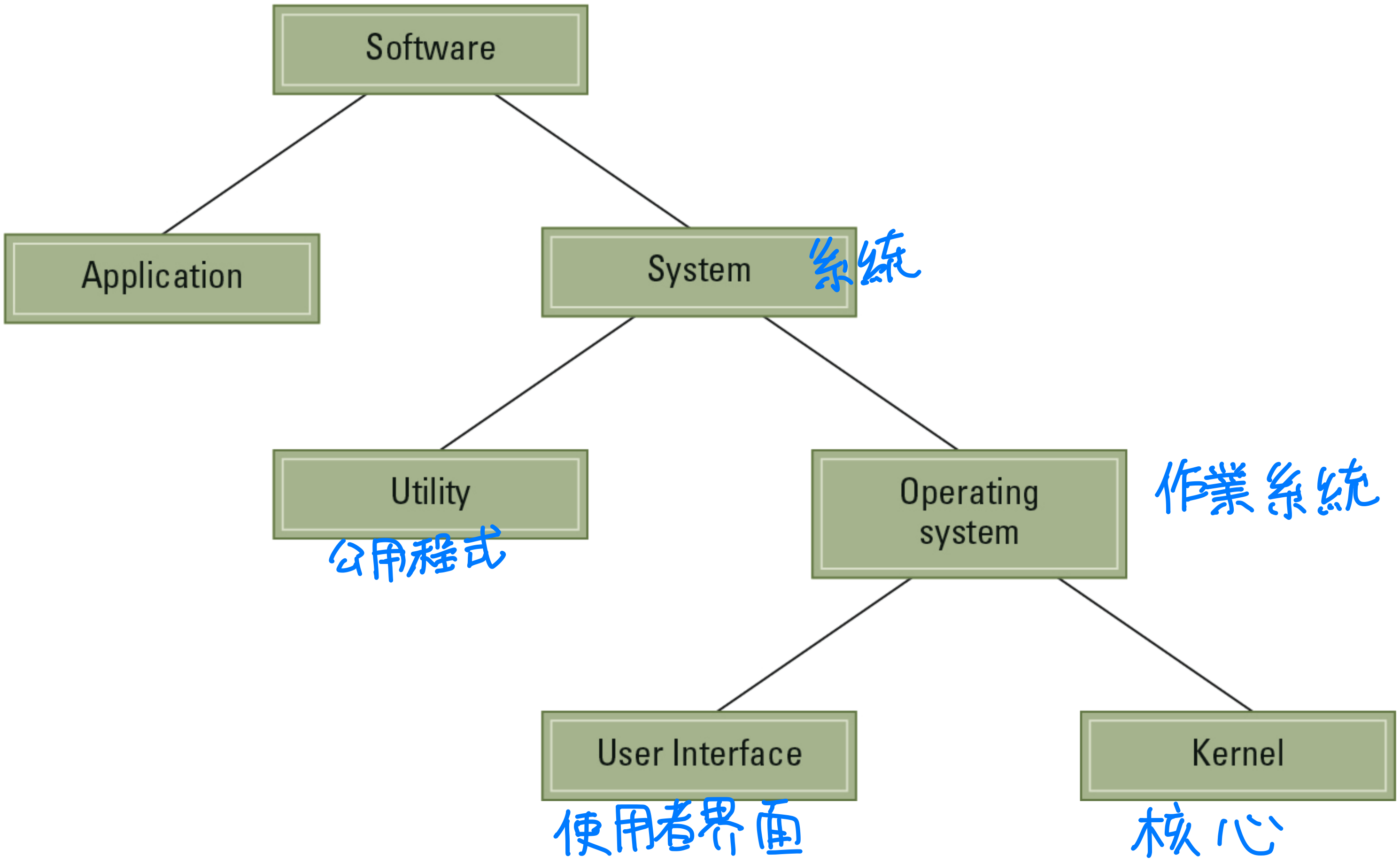


3.2 Operating System Architecture

- Application software 應用軟体
  - Performs specific tasks for users (productivity, games, software development)
- System software 系統軟体
  - Provides infrastructure for application software
  - Consists of operating system and utility software 公用程式

Pearson

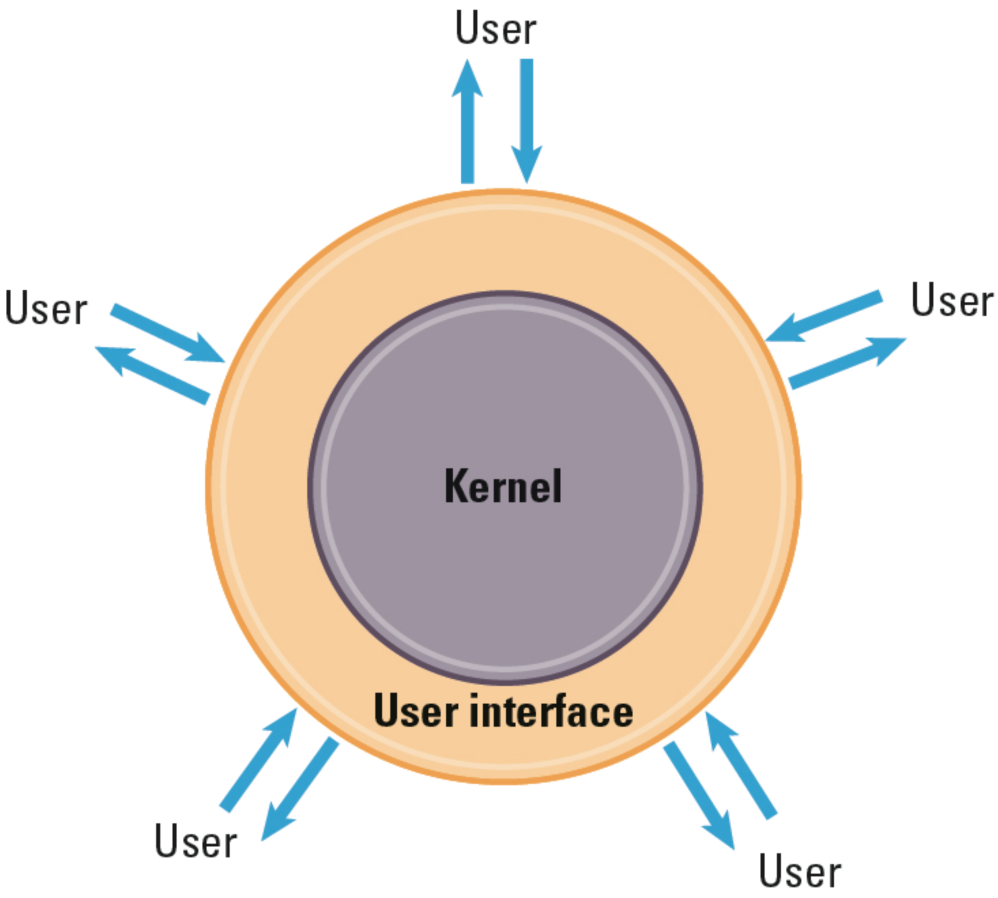
Figure 3.3 Software classification



Operating System Components

- **User Interface:** Communicates with users 使用者界面
  - Text based (Shell)
  - Graphical user interface (GUI)
- **Kernel:** Performs basic required functions
  - File manager 檔案管理
  - Device drivers 裝置驅动
  - Memory manager 记忆体管理
  - Scheduler and dispatcher 排程器& 调度器

Figure 3.4 The user interface acts as an intermediary between users and the operating system’s kernel



File Manager

目錄 资料夾

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

路徑 子目錄

- **Directory Path:** A sequence of directories within directories

Pearson

Memory Manager 记忆体管理

- Allocates space in main memory

分配

- 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

虚拟记忆体

Page的大小同時也會影響

记忆体的效能

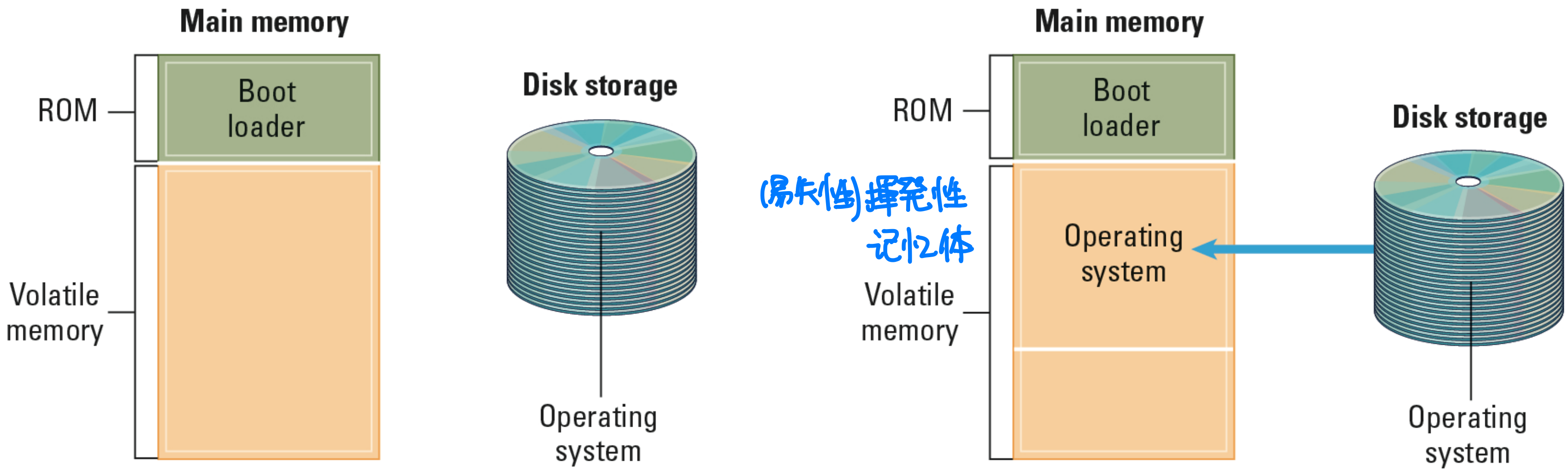
Getting it Started (Bootstrapping)

載入引導作業系統

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

Pearson

Figure 3.5 The booting process 啟动程序



(易失性)揮発性
记忆体

Step 1: Machine starts by executing the boot loader program already in memory. Operating system is stored in mass storage.

Step 2: Boot loader program directs the transfer of the operating system into main memory and then transfers control to it.

Pearson

3.3 Coordinating the Machine's Activities

协 调

An operating system coordinates the execution of application software, utility software, and units within the operating system itself.

The Concept of a Process

執行緒(進程)

- **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

排程器

- **Scheduler:** Adds new processes to the process table and removes completed processes from the process table

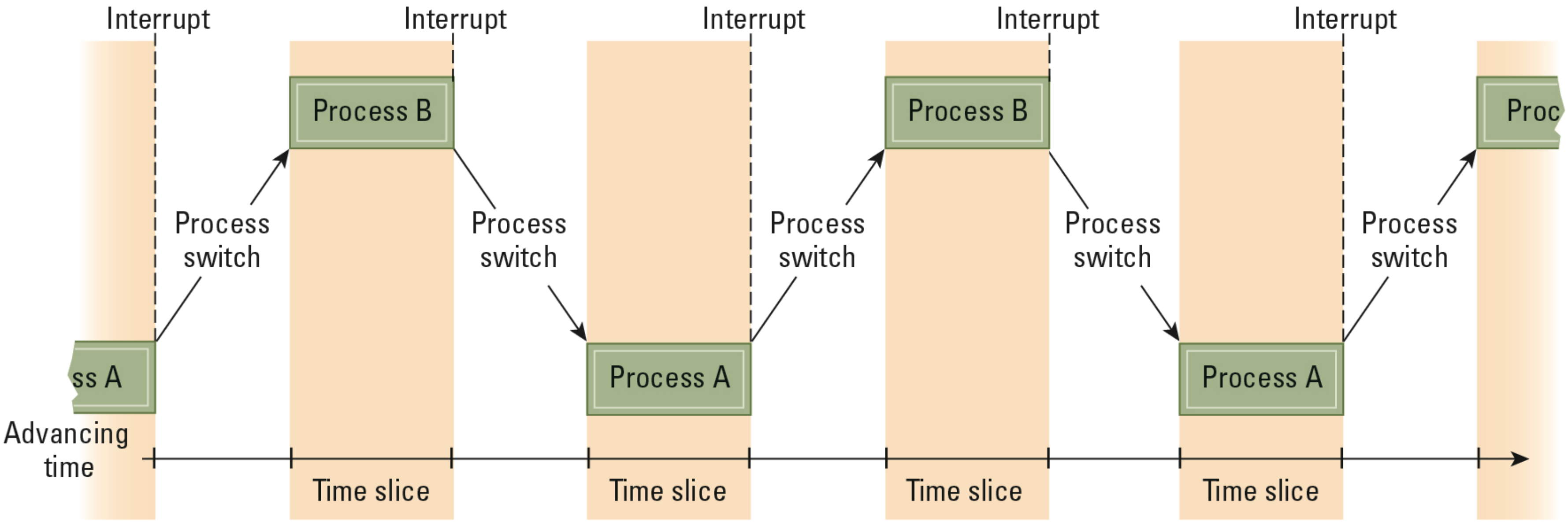
调度器 (類似计時器, 控制 Time Slice)

- **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.

Pearson

Figure 3.6 Multiprogramming between process A and process B

Time-sharing 分時系統



執行緒切換 P.179
Process Switch (Context Switch)

3.4 Handling Competition Among Processes

號誌 控制標誌 Process 間競爭的處理

- **Semaphore:** A “control flag”

(Critical Section 臨界區域、臨界區段)

- **Critical Region:** A group of instructions that should be executed by only one process at a time

互斥鎖

- **Mutual exclusion:** Requirement that only one process at a time be allowed to execute a Critical Region

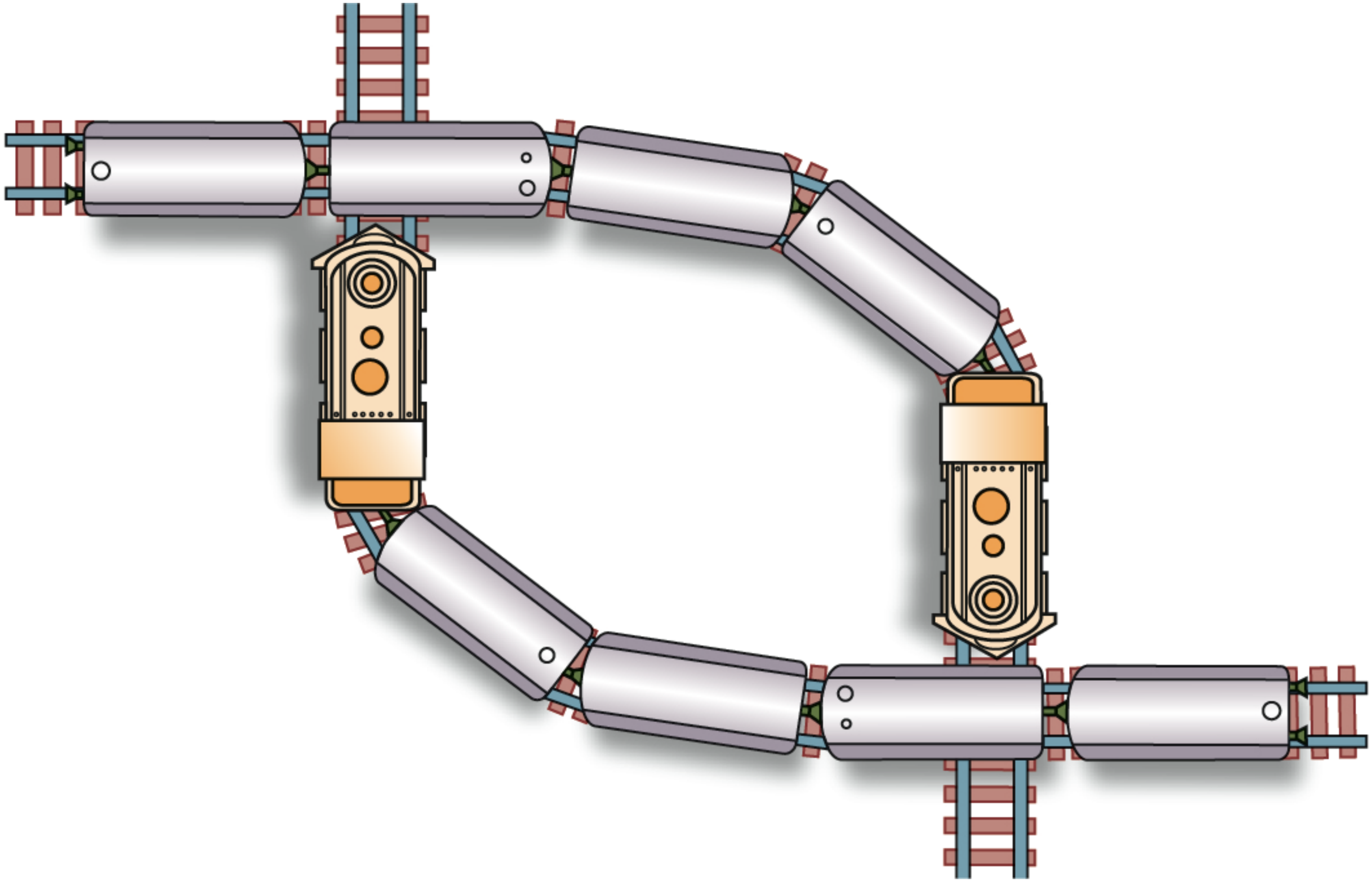
Deadlock 死結、死鎖

、process 間彼此相互箝制

- Processes block each other from continuing because each is waiting for a resource that is allocated to another
- Conditions required for deadlock 死結發生の要件
  1. Competition for non-sharable resources 競爭的资源非共享
  2. Resources requested on a partial basis 在已經有资源的基礎上要求更多资源。
  3. An allocated resource can not be forcibly retrieved

     已被分配給Process之资源不能被強制回收(強制釋放)

Figure 3.7 A deadlock resulting from competition for nonshareable railroad intersections



3.5 Security p.187

- Attacks from outside
  - Problems
    - Insecure passwords 弱密碼
    - Sniffing software 數據包分析器 (嗅探軟体)
  - Counter measures
    - Auditing software 稽核軟体、技術 (e.g. 防毒軟体、防火牆)

Security (continued)

- Attacks from within 來自內部的攻擊
  - Problem: A process that gains access to memory outside its designated area 存取其指定區域之外的记忆体進程
  - Counter measures: Control process activities via privilege levels and privileged instructions

特權模式

透過特權模式 or 特權指令
控制進程活动状態