

# List of Sidebars

## Part I

---

### CHAPTER 1 Systems

Sidebar 1.1: Stopping a Supertanker .....	6
Sidebar 1.2: Why Airplanes can't Fly .....	6
Sidebar 1.3: Terminology: Words Used to Describe System Composition .....	9
Sidebar 1.4: The Cast of Characters and Organizations.....	14
Sidebar 1.5: How Modularity Reshaped the Computer Industry .....	21
Sidebar 1.6: Why Computer Technology has Improved Exponentially with Time .....	32

### CHAPTER 2 Elements of Computer System Organization

Sidebar 2.1: Terminology: Durability, Stability, and Persistence .....	46
Sidebar 2.2: How Magnetic Disks Work.....	49
Sidebar 2.3: Representation: Pseudocode and Messages .....	54
Sidebar 2.4: What is an Operating System? .....	79
Sidebar 2.5: Human Engineering and the Principle of Least Astonishment .....	85

### CHAPTER 3 The Design of Naming Schemes

Sidebar 3.1: Generating a Unique Name from a Timestamp.....	125
Sidebar 3.2: Hypertext Links in the Shakespeare Electronic Archive.....	129

### CHAPTER 4 Enforcing Modularity with Clients and Services

Sidebar 4.1: Enforcing Modularity with a High-Level Languages.....	154
Sidebar 4.2: Representation: Timing Diagrams .....	156
Sidebar 4.3: Representation: Big-Endian or Little-Endian?.....	158
Sidebar 4.4: The X Window System.....	162
Sidebar 4.5: Peer-to-peer: Computing without Trusted Intermediaries.....	164

### CHAPTER 5 Enforcing Modularity with Virtualization

Sidebar 5.1: RSM, Test-and-Set and Avoiding Locks.....	224
Sidebar 5.2: Constructing a Before-or-After Action without Special Instructions .....	226
Sidebar 5.3: Bootstrapping an Operating System .....	239
Sidebar 5.4: Process, Thread, and Address Space .....	249
Sidebar 5.5: Position-Independent Programs.....	251

<b>Sidebar 5.6:</b> Interrupts, Exceptions, Faults, Traps, and Signals.....	259
<b>Sidebar 5.7:</b> Avoiding the Lost Notification Problem with Semaphores .....	277

## **CHAPTER 6 Performance**

<b>Sidebar 6.1:</b> Design Hint: When in Doubt Use Brute Force .....	301
<b>Sidebar 6.2:</b> Design Hint: Optimize for the Common Case .....	307
<b>Sidebar 6.3:</b> Design Hint: Instead of Reducing Latency, Hide It.....	310
<b>Sidebar 6.4:</b> RAM Latency .....	323
<b>Sidebar 6.5:</b> Design Hint: Separate Mechanism from Policy .....	330
<b>Sidebar 6.6:</b> OPT is a Stack Algorithm and Optimal.....	343
<b>Sidebar 6.7:</b> Receive Livelock .....	350
<b>Sidebar 6.8:</b> Priority Inversion.....	358

## **PART II [ON-LINE]**

---

### **CHAPTER 7 The Network as a System and a System Component**

<b>Sidebar 7.1:</b> Error Detection, Checksums, and Witnesses
<b>Sidebar 7.2:</b> The Internet
<b>Sidebar 7.3:</b> Framing Phase-Encoded Bits
<b>Sidebar 7.4:</b> Shannon's Capacity Theorem
<b>Sidebar 7.5:</b> Other End-to-End Transport Protocol Interfaces
<b>Sidebar 7.6:</b> Exponentially Weighted Moving Averages
<b>Sidebar 7.7:</b> What does an Acknowledgment Really Mean?
<b>Sidebar 7.8:</b> The Tragedy of the Commons
<b>Sidebar 7.9:</b> Retrofitting TCP
<b>Sidebar 7.10:</b> The Invisible Hand

### **CHAPTER 8 Fault Tolerance: Reliable Systems from Unreliable Components**

<b>Sidebar 8.1:</b> Reliability Functions
<b>Sidebar 8.2:</b> Risks of Manipulating MTTFs
<b>Sidebar 8.3:</b> Are Disk System Checksums a Wasted Effort?
<b>Sidebar 8.4:</b> Detecting Failures with Heartbeats.

### **CHAPTER 9 Atomicity: All-or-nothing and Before-or-after**

<b>Sidebar 9.1:</b> Actions and Transactions
<b>Sidebar 9.2:</b> Events that Might Lead to Invoking an Exception Handler
<b>Sidebar 9.3:</b> Cascaded Aborts
<b>Sidebar 9.4:</b> The Many Uses of Logs

### **CHAPTER 10 Consistency**

**CHAPTER 11 Information Security**

**Sidebar 11.1:** Privacy

**Sidebar 11.2:** Should Designs and Vulnerabilities be Public?

**Sidebar 11.3:** Malware: Viruses, Worms, Trojan Horses, Logic Bombs, Bots, Drive-by Downloads, etc.

**Sidebar 11.4:** Why are Buffer Overrun Bugs so Common?

**Sidebar 11.5:** Authenticating Personal Devices: the Resurrecting Duckling Policy

**Sidebar 11.6:** The Kerberos Authentication System

**Sidebar 11.7:** Economics of Computer Security

**Sidebar 11.8:** Secure Hash Algorithm (SHA)