

1. Record Nr.	UNINA9910455028603321
Autore	Saltzer J. H. <1939->
Titolo	Principles of computer system design [[electronic resource]] : an introduction / / Jerome H. Saltzer, M. Frans Kaashoek
Pubbl/distr/stampa	Amsterdam ; ; London, : Elsevier/Morgan Kaufmann, c2009
ISBN	1-282-25834-6 9786612258343 0-08-095942-3
Edizione	[1st edition]
Descrizione fisica	1 online resource (561 p.)
Altri autori (Persone)	KaashoekFrans <1965->
Disciplina	005.1
Soggetti	Computers - Design Computer architecture Computer organization Software engineering Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Front Cover; Half Title Page; Title Page; Copyright Page; Dedication Page; Table of Contents; List of Sidebars; Preface; Where to find Part II and other On-line Materials; Acknowledgments; Chapter 1. Systems; Overview; 1.1 Systems and Complexity; 1.1.1 Common Problems of Systems in Many Fields; 1.1.2 Systems, Components, Interfaces, and Environments; 1.1.3 Complexity; 1.2 Sources of Complexity; 1.2.1 Cascading and Interacting Requirements; 1.2.2 Maintaining High Utilization; 1.3 Coping with Complexity I; 1.3.1 Modularity; 1.3.2 Abstraction; 1.3.3 Layering; 1.3.4 Hierarchy 1.3.5 Putting it Back Together: Names make Connections1.4 Computer Systems are the Same but Different; 1.4.1 Computer Systems have no Nearby Bounds on Composition; 1.4.2 d(technology)/dt is Unprecedented; 1.5 Coping with Complexity II; 1.5.1 Why Modularity, Abstraction, Layering, and Hierarchy aren't Enough; 1.5.2 Iteration; 1.5.3 Keep it Simple; What the Rest of this Book is About; Exercises; Chapter 2. Elements of Computer System Organization; Overview; 2.1 The Three Fundamental Abstractions; 2.1.1 Memory; 2.1.2 Interpreters;

2.1.3 Communication Links; 2.2 Naming in Computer Systems
2.2.1 The Naming Model2.2.2 Default and Explicit Context References;
2.2.3 Path Names, Naming Networks, and Recursive Name Resolution;
2.2.4 Multiple Lookup: Searching through Layered Contexts; 2.2.5
Comparing Names; 2.2.6 Name Discovery; 2.3 Organizing Computer
Systems with Names and Layers; 2.3.1 A Hardware Layer: The Bus;
2.3.2 A Software Layer: The File Abstraction; 2.4 Looking Back and
Ahead; 2.5 Case Study: UNIX® File System Layering and Naming; 2.5.1
Application Programming Interface for the UNIX File System; 2.5.2 The
Block Layer; 2.5.3 The File Layer; 2.5.4 The Inode Number Layer
2.5.5 The File Name Layer2.5.6 The Path Name Layer; 2.5.7 Links; 2.5.8
Renaming; 2.5.9 The Absolute Path Name Layer; 2.5.10 The Symbolic
Link Layer; 2.5.11 Implementing the File System API; 2.5.12 The Shell
and Implied Contexts, Search Paths, and Name Discovery; 2.5.13
Suggestions for Further Reading; Exercises; Chapter 3. The Design of
Naming Schemes; Overview; 3.1 Considerations in the Design of
Naming Schemes; 3.1.1 Modular Sharing; 3.1.2 Metadata and Name
Overloading; 3.1.3 Addresses: Names that Locate Objects; 3.1.4
Generating Unique Names; 3.1.5 Intended Audience and User-Friendly
Names
3.1.6 Relative Lifetimes of Names, Values, and Bindings3.1.7 Looking
Back and Ahead: Names are a Basic System Component; 3.2 Case Study:
The Uniform Resource Locator (URL); 3.2.1 Surfing as a Referential
Experience; Name Discovery; 3.2.2 Interpretation of the URL; 3.2.3 URL
Case Sensitivity; 3.2.4 Wrong Context References for a Partial URL;
3.2.5 Overloading of Names in URLs; 3.3 War Stories: Pathologies in the
Use of Names; 3.3.1 A Name Collision Eliminates Smiling Faces; 3.3.2
Fragile Names from Overloading, and a Market Solution
3.3.3 More Fragile Names from Overloading, with Market Disruption

Sommario/riassunto

This text identifies, examines, and illustrates fundamental concepts in computer system design that are common across operating systems, networks, database systems, distributed systems, programming languages, software engineering, security, fault tolerance, and architecture. Through carefully analyzed case studies from each of these disciplines, it demonstrates how to apply these concepts to tackle practical system design problems. To support the focus on design, the text identifies and explains abstractions that have proven successful in practice such as, remote procedure call, client
