1. Record Nr. UNINA9910366591603321

Autore	Schagaev Igor
Titolo	Software Design for Resilient Computer Systems / / by Igor Schagaev, Eugene Zouev, Kaegi Thomas
Pubbl/distr/stampa	Cham : , : Springer International Publishing : , : Imprint : Springer, , 2020
ISBN	3-030-21244-0
Edizione	[2nd ed. 2020.]
Descrizione fisica	1 online resource (xiv, 214 pages) : illustrations
Disciplina	005
Soggetti	Electrical engineering Electronic circuits Software engineering Computer software—Reusability Quality control Reliability Industrial safety Communications Engineering, Networks Circuits and Systems Software Engineering Performance and Reliability Quality Control, Reliability, Safety and Risk
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Introduction -- Hardware Faults -- Fault Tolerance: Theory and Concepts -- Generalized Algorithm of Fault Tolerance (GAFT) -- GAFT Generalization: A Principle and Model of Active System Safety -- System Software Support for Hardware Deficiency: Function and Features -- Testing and Checking -- Recovery Preparation -- Recovery: Searching and Monitoring of Correct Software States -- Recovery Algorithms: An Analysis -- Programming Language for Safety Critical Systems -- Proposed Runtime System Structure -- Proposed Runtime System vs. Existing Approaches -- Hardware: The ERRIC Architecture -- Architecture Comparison and Evaluation -- Reliability of ERRIC --

Performance of ERRIC -- ERRIC Software -- How about resilience at large -- Map of Resilience.

Sommario/riassunto

This book addresses the question of how system software should be designed to account for faults, and which fault tolerance features it should provide for highest reliability. With this second edition of Software Design for Resilient Computer Systems the book is thoroughly updated to contain the newest advice regarding software resilience. With additional chapters on computer system performance and system resilience, as well as online resources, the new edition is ideal for researchers and industry professionals. The authors first show how the system software interacts with the hardware to tolerate faults. They analyze and further develop the theory of fault tolerance to understand the different ways to increase the reliability of a system, with special attention on the role of system software in this process. They further develop the general algorithm of fault tolerance (GAFT) with its three main processes: hardware checking, preparation for recovery, and the recovery procedure. For each of the three processes, they analyze the requirements and properties theoretically and give possible implementation scenarios and system software support required. Based on the theoretical results, the authors derive an Oberon-based programming language with direct support of the three processes of GAFT. In the last part of this book, they introduce a simulator, using it as a proof of concept implementation of a novel fault tolerant processor architecture (ERRIC) and its newly developed runtime system feature-wise and performance-wise. Due to the wide reaching nature of the content, this book applies to a host of industries and research areas, including military, aviation, intensive health care, industrial control, and space exploration.