

Synchronization in Real-Time Systems

A Priority Inheritance Approach

Real-time computing systems are vital to a wide range of applications. For example, they are used in the control of nuclear reactors and automated manufacturing facilities, in controlling and tracking air traffic, and in communication systems. In recent years, real-time systems have also grown larger and become more critical. For instance, advanced aircraft such as the space shuttle must depend heavily on computer systems [Carlow 84]. The centralized control of manufacturing facilities and assembly plants operated by robots are other examples at the heart of which lie embedded real-time systems. Military defense systems deployed in the air, on the ocean surface, land and underwater, have also been increasingly relying upon real-time systems for monitoring and operational safety purposes, and for retaliatory and containment measures. In telecommunications and in multi-media applications, real time characteristics are essential to maintain the integrity of transmitted data, audio and video signals. Many of these systems control, monitor or perform critical operations, and must respond quickly to emergency events in a wide range of embedded applications. They are therefore required to process tasks with stringent timing requirements and must perform these tasks in a way that these timing requirements are guaranteed to be met. Real-time scheduling algorithms attempt to ensure that system timing behavior meets its specifications, but typically assume that tasks do not share logical or physical resources. Since resource-sharing cannot be eliminated, synchronization primitives must be used to ensure that resource consistency constraints are not violated.

Real-time computing systems are vital to a wide range of applications. For example, they are used in the control of nuclear reactors and automated manufacturing facilities, in controlling and tracking air traffic, and in communication systems. In recent years, real-time systems have also grown larger and become more critical. For instance, advanced aircraft such as the space shuttle must depend heavily on computer systems [Carlow 84]. The centralized control of manufacturing facilities and assembly plants operated by robots are other examples at the heart of which lie embedded real-time systems. Military defense systems deployed in the air, on the ocean surface, land and underwater, have also been increasingly relying upon real-time systems for monitoring and operational safety purposes, and for retaliatory and containment measures. In telecommunications and in multi-media applications, real time characteristics are essential to maintain the integrity of transmitted data, audio and video signals. Many of these systems control, monitor or perform critical operations, and must respond quickly to emergency events in a wide range of embedded applications. They are therefore required to process tasks with stringent timing requirements and must perform these tasks in a way that these timing requirements are guaranteed to be met. Real-time scheduling algorithms attempt to ensure that system timing behavior meets its specifications, but typically assume that tasks do not share logical or physical resources. Since resource-sharing cannot be eliminated, synchronization primitives must be used to ensure that resource consistency constraints are not violated.

SYNCHRONIZATION
IN REAL-TIME
SYSTEMS
A Priority Inheritance
Approach

Ragunathan Rajkumar

KLUWER ACADEMIC PUBLISHERS

106,99 €

99,99 € (zzgl. MwSt.)

Lieferfrist: bis zu 10 Tage

Artikelnummer: 9780792392118

Medium: Buch

ISBN: 978-0-7923-9211-8

Verlag: Humana

Erscheinungstermin: 31.08.1991

Sprache(n): Englisch

Auflage: 1. Auflage 1991

Serie: The Springer International Series in Engineering and Computer Science

Produktform: Gebunden

Gewicht: 1010 g

Seiten: 176

Format (B x H): 160 x 241 mm

