Real-Time Operating Systems: Can Theoretical Solutions Match with Practical Needs

Helmut Rzehak

University of the Federal Armed Forces Munich
Werner-Heisenberg-Weg 39
85579 Neubiberg
Germany

Abstract

Constructing real-time systems, which are predictable in a very restrictive sense, is a challenging task for scientists. On the other hand today’s real-time operating systems do not meet these strong requirements. This paper gives an insight into sources of delays for application processes caused by the operating system. It follows from this analysis that most of the services of real-time operating systems insert unexpected delays to the application processes and worst case values are hardly to determine. Regarding the fact that real-time operating systems are used successfully certain deviations from the model of strict predictability seem to be tolerable for most applications. The paper presents some ideas to describe such tolerable deviations more precisely.

Keywords

Real-time operating systems, latency times, real-time performance metrics, concurrency control.

1 Introduction

A lot of papers has been published in the scientific community concerning real-time operating systems in general as well as special aspects about these. On the other hand off-the-shelf real-time operating systems are available and interface definitions are going to be standardized. At a first glance the various products differ in the scope of applications they support but within the same scope they look very similar. Much of the scientific work has influenced just a little bit the established solutions. The approved “good practice” as a prerequisite for standardisation seems to be somehow different from results achieved by scientific works.
This mismatch may simply come from the ignorance of some handy "experts" or may have some serious reasons. This paper tries to give an answer to this question taking into account some different application aspects of real-time operating systems.

2 Concepts for Real-Time Operating Systems

2.1 Critical View of some Definitions

The central point of most definitions for real-time computing is a required deterministic (or predictable) behaviour in the time domain. In contrast with this really existing systems contain many sources of nondeterminism. Among others we have arbitrary sequences of events to react on and generally it is hard to define how a system should react in a deterministic way on non-predictable input sequences. At the occurrence of the event the whole internal state of the system is usually unknown and this results in a non-predictable sequence of state transitions for performing the desired action. Having a close look to reality the very task in designing a real-time system is to reduce nondeterminisms in the overall system behaviour under an acceptable level because the complete elimination is impossible or at least too expensive. Basicly this means to describe the term "predictability" not by a binary value but by a fuzzy set using the membership function to determine if the level of predictability is sufficient. The conventional thinking about hard deadlines and mathematical proof of certain behavioural attributes is included as a special case. This new understanding of predictability gives a more realistic view of at least three aspects of real-time systems without leaving a solid basis for reasoning:

1. Certain deviations of the intended system behaviour may be acceptable if they occur very rarely. This is a realistic assumption for many (not to say for most) applications. Frequently worst case considerations are far to pessimistic and lead to oversized systems. E.g. an upper bound for the latency times introduced by the operating system, which will be discussed later on in this paper, may be considerably high, but the worst case situation will occur only under very special conditions. Perhaps these conditions will never be valid for an application.

2. Instead of describing the time constraints of a real-time task only by its deadline we can use a benefit function like in the Benefit Accrual Model proposed in the Alpha real-time operating system ([11], [12]). As a next step we can replace a certain benefit function by a fuzzy set of functions as shown in fig. 1.

3. We should take into account the limited resolution of the time scale. This is not due to the limited frequency of the clock ticks but due to arbitrary delays for event and message handling by the different subsystems. For distributed applications this aspect is particularly important ([25], [26]).

Our proposal is not covered by the classic distinction of hard and soft real-time systems. There exist only "soft" descriptions what a soft real-time system is. This