Sharif University of Technology
Department of Computer Science and Engineering

Lec. 0:
Introduction

Real-Time Computing

S. Safari
Fall 2023

Definition

- Real-time systems are computing systems that must react within precise time constraints to events in the environment.
- The correct behavior of these systems depends:
  - The value of the computation
  - On the time at which the results are produced

- A reaction that occurs too late could be useless or even dangerous.
- Typical misconception:
  - Real-time computing ≠ compute things as fast as possible
  - Real-time computing = compute as fast as necessary, but not too fast

Applications of Real-Time Computing

- Chemical and nuclear plant control
- Control of complex production processes
- Railway switching systems
- Automotive applications
- Flight control systems
- Environmental acquisition and monitoring
- Telecommunication systems
- Medical systems
- Industrial automation
- Robotics

Applications of Real-Time Computing (2)

- Military systems
- Space missions
- Consumer electronic devices
- Multimedia systems
- Smart toys
- Virtual reality

Real-Time Computers

- In many cases, the real-time computer running the application is embedded into the system to be controlled.
- Embedded systems span from small portable devices (e.g., cellular phones, cameras, navigators, smart toys) to larger systems (e.g., industrial robots, cars, aircrafts).

Different Approaches for Designing Real-Time Systems

- Ad-hoc techniques and heuristic approaches
  - Not suitable
- Writing large portions of code in assembly language, programming timers, writing low-level drivers for device handling, and manipulating task and interrupt priorities
  - These techniques can be optimized to run very efficiently, but have the following disadvantages:
    - Tedious programming
    - Difficult code understanding
    - Difficult software maintainability
    - Difficult verification of time constraints

Using Empirical Techniques for RT Systems

- Highly unpredictable
- High failure rate
- If all critical time constraints cannot be verified a priori and the operating system does not include specific mechanisms for handling real-time tasks, the system could apparently work well for a period of time, but it could collapse in certain rare, but possible, situations.
  - Catastrophic and may injure people or cause serious damages to the environment

A Failure Example: Space Shuttle

- The first flight of the space shuttle was delayed, at considerable cost, because of a timing bug that arose from a transient overload during system initialization on one of the redundant processors dedicated to the control of the aircraft.
- Although the shuttle control system was intensively tested, the timing error was not discovered. Later, by analyzing the code of the processes, it was found that there was only a 1 in 67 probability (about 1.5 percent) that a transient overload during initialization could push the redundant processor out of synchronization.

Another Failure Example: Patriot Missiles

- Another software bug was discovered on the real-time control system of the Patriot missiles, used to protect Saudi Arabia during the Gulf War.
- On February 25, 1991, the radar sighted a Scud missile directed at Saudi Arabia, and the onboard computer predicted its trajectory, performed the verification, but classified the event as a false alarm.
- Causing injuries and enormous economic damage.
- Because of a long interrupt handling routine running with disable interrupts, the real-time clock of the onboard computer was missing some clock interrupts, thus accumulating a delay of about 57 microseconds per minute.
- Such a delay caused a prediction error in the verification phase of 687 meters!

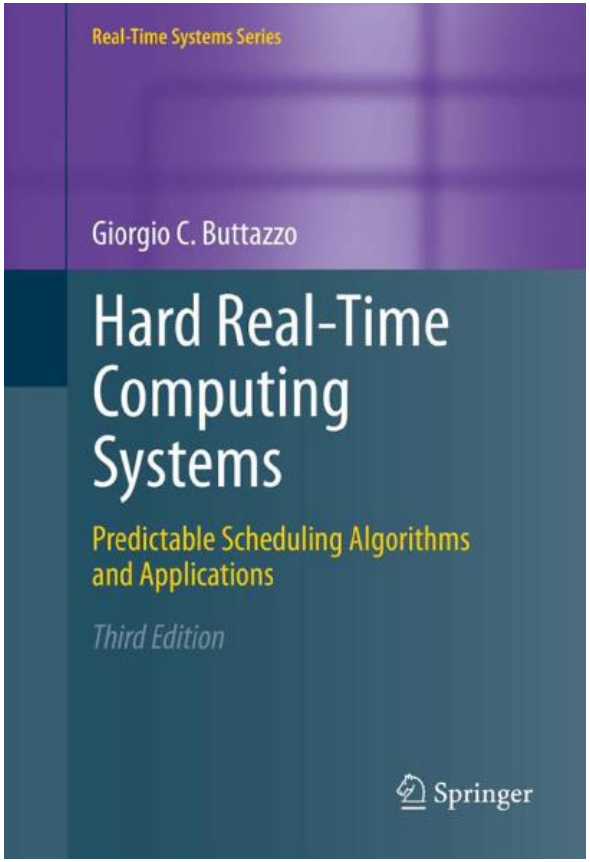
Predictability in Real-Time Systems

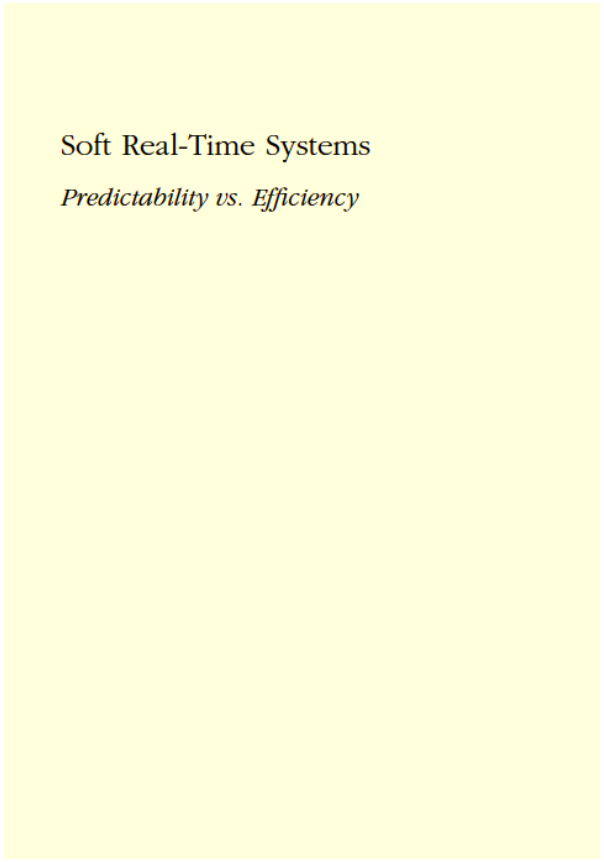
- Software testing, although important, does not represent a solution for achieving predictability in real-time systems.
- The program flow depends on input sensory data and environmental conditions, which cannot be fully replicated during the testing phase.
  - A partial verification of the software behavior, relative to the particular subset of data provided as input.
- Achieving a more robust guarantee of the performance of a real-time system
  - Using more sophisticated design methodologies, combined with a static analysis of the source code and specific operating systems mechanisms, purposely designed to support computation under timing constraints.

Broad set of topics

1. Introduction
2. Basic Concepts
3. Aperiodic and Periodic Task Scheduling
4. Fixed- and Dynamic Priority Servers
5. Resource Access Protocols
6. Limited Preemptive Scheduling
7. Real-Time Operating Systems and Standards
8. Task Mapping and Scheduling on Multiprocessor Systems
9. Mixed-Criticality Systems

Textbooks


Real-Time Systems Series
Giorgio C. Buttazzo
Hard Real-Time Computing Systems
Predictable Scheduling Algorithms and Applications
Third Edition
Springer


Soft Real-Time Systems
Predictability vs. Efficiency

Summary

- Definition of Real-Time Systems
- Applications of Real-Time Systems
- Designing of Real-Time Systems
- Example of failed RTSs
- The topics of course
- References

Grading

- Midterm Exam: 25%-35%
- Quizzes : 5%-10%
- Homework: 10%-15%
- Project: 20%-25%
- Final Exam: 35%-45%