I.T.U.

Faculty of Computer and Informatics

Computer Engineering

Lesson name: Real-Time Systems Software

Lesson Code: BLG 450E

Name Surname: Abdullah AYDEĞER

Number: 040090533

Instructor’s Name: Turgay ALTILAR

Due Date: 17.02.2013

Misconceptions About Real-Time Computing

In computer science there is one research topic about real-time systems and their software. But there are lots of misconceptions about this area. Before mentioning about these misconceptions, definition about real-time computing will be given.

What is Real-Time Computing?

Every system which uses computer should generate results logically (as our expectations). But most of the systems aren't designed with considering time necessities. Real-time computing is required to think time constraints too. Today real-time system is used lots of such as flight control systems.

Next-generation real-time systems will require artificial intelligence and rapid advance in hardware. Therefore it is current research topic and scientists should focus this topic.

Common Misconceptions

Computer scientists are not interested in real-time systems. This caused some misconceptions on this area. Some of them will be explained follow:

I. There is no science in real-time system design

Real-time system is designed ad-hocly is a fact but this does not mean scientific approach is not possible.

II. Advances in supercomputer hardware will take care of real-time requirements

Using parallel processors for improving system throughput will not meet the time requirements of system. In fact, it can make the system worse.

III. Real-time computing is equivalent to fast computing

Fast computing handles to get minimum time for some set of tasks. On the other hand, real-time computing is focused each task's time requirement individually. Most important feature of real-time system is predictability that means functional and timing behavior should be deterministic as necessary.

IV. Real-time programming is assembly coding, priority interrupt programming, and device driver writing

Real-time programming is required to do machine-level optimization.

V. Real-time research is performance engineering

Real-time systems are generally designed with effective resource allocation to satisfy system requirements.

VI. The problem in real-time system design have all been solved in other areas of computer science or operations research

There are some unique problems of real-time systems which are indeed different than other research areas.

VII. It is meaningful to talk about guaranteeing real-time performance, because we can not guarantee that the hardware will not fail and software is bug free or that the actual operating conditions will not violate the specified design limits

We can not guarantee anything outside over control, but what we can guarantee, we should.

VIII. Real-time systems function in a static environment

Real-time system may have to satisfy different sets of timing constraints at different times.

The challenge of real-time computing systems

I. Specification and verification:

Fundamental challenge in specification and verification is incorporating time metric in the real-time system. If time is not concerned, design is generally simple. This should be changed. For example, fairness is not proper for real-time systems.

II. Real-time scheduling theory:

Scheduling theory addresses the problem of meeting the specified timing requirements. Scheduling theory is used for other areas than real-time systems. But they use different methods to do it. Real-time system is often highly dynamic, requiring online, adaptive scheduling algorithms. Such algorithms must be based on heuristics, since these scheduling problems are NP-hard. In these cases the goal is to schedule as many as possible tasks with considering timing constraints. Alternative ones are required to be integrated online scheduler.

There are also some other measures too, such as penalty functions defined missed(deadline miss) tasks, weighted success ratio which is the percentage of tasks that meet their deadlines weighted by importance of those tasks.

III. Real-time operating systems:

The operating system must provide basic support for guaranteeing real-time constraints, supporting fault-tolerance and distribution, and integrating time-constrained resource allocations and scheduling across a spectrum of resource types. For example, if the tasks are cooperated in the system, operating system should be highly dynamic.

IV. Real-time programming languages and design methodology:

Programming abstractions is required to design and implement complex real-time systems. Programming languages should have followings:

- Support for the management of time
- Schedulability check
- Reusable real-time software modules
- Support for distributed programs and fault tolerance

V. Distributed real-time databases:

Real-time databases is also one type of real-time systems and have some constraints as data consistency, transaction correctness and transaction deadlines.

VI. Artificial intelligence:

Real-time AI research emphasizes reasoning about time-constrained processes and using heuristic knowledge to control or schedule these processes.

VII. Real-time system architectures:

Many real-time systems can be viewed as three stages: data acquisition from sensors, data processing and output to actuators and/or displays. On the other hand, specified architecture should change with the application.

VIII. Real-time communication:

Communication is a real-time application and it should be stable. Following research should be developed to support unique challenges in real-time communications:

- Dynamic routing solutions with guaranteed timing correctness
- Network buffer management that support scheduling solutions
- Fault-tolerance and time-constrained communications
- Network scheduling that can be combined with processor scheduling to provide system-level scheduling solutions.

Conclusion

Real-time systems will be much more complex than what they are today. They will work in distributed and highly dynamic environments. This will cause complex timing constraints which are obligatorily needed to be met.