Hierarchical Real-Time Scheduling

Assignment 2
Research Planning, 2008-2009

Mikael Åsberg mikael.asberg@mdh.se

February 9, 2009
Contents

1 My research area 3
  1.1 Introduction .............................................. 3
  1.2 Ongoing work .............................................. 4
  1.3 Research method .......................................... 4

2 Current research issues and hot topics 5

3 Central literature 5

4 Key conferences and leading research groups 6
  4.1 Key conferences ........................................... 6
  4.2 Leading research groups .................................. 7

5 Relation and relevance for own research 7
1 My research area

1.1 Introduction

Hard real-time applications have strict timing properties which most often are mathematically analyzed, based on the system properties, and thereby known to be schedulable or not. The schedulability of a system is strictly dependent on the system properties and these should not be altered in order for the system to remain schedulable.

However, integrating different real-time systems on the same computer is positive due to a potential cost reduction inherent in lowering the amount of hardware needed [3], less Central Processing Unit’s (CPU) are required. This motivation comes from the fact that CPU’s are getting faster.

Another motivation for integrating real-time applications is the component technology [4]. Component-based design is a well known design paradigm which makes it possible to decompose large complex systems into smaller parts, thereby reducing complexity. Also, it facilitates reusability which is good for development cost reduction.

Execution of different real-time applications on a single processor system have great impact on the timing requirements of each application. These application’s timing requirements are dependent on the amount of resources (for example CPU) available, in order for the application to be schedulable. This makes it difficult to share resources among real-time systems while at the same time preserving temporal behavior [3]. It might be very difficult, time consuming or in the worst case impossible to adopt real-time software to new environments where it must share resources with other applications. Merging several real-time systems together, sharing a CPU, is a big challenge. All systems must have their properties re-configure and re-analyzed to be able to execute together.

We see that it is a big challenge to let real-time applications share resources with each other. A solution to this problem is to introduce hierarchical scheduling. The benefits are the following:

- Integration is easy, efficient and analyzable.
- Temporal behavior of each real-time application is preserved.
- No need to re-configure the real-time applications that are being integrated.
1.2 Ongoing work

During my master project I implemented a two-level Hierarchical Scheduling Framework (HSF) in VxWorks [6].

In long term, I will investigate the possibility of using the HSF in multi-core.

In short term, interesting projects are case studies of my implemented two-level scheduler and implementation/evaluation of existing resource sharing protocols for hierarchical systems.

Related to my visit to NXP (Eindhoven), there are many interesting topics. Firstly, explore the possibilities to implement a HSF in General Purpose Operating System (GPOS) Linux. Due to my recent visit to RTSS (Barcelona) I’ve started some collaborations with Japanese and Italian PhD students/researchers in the field of scheduling in Linux. Secondly, look at online (for scheduling decisions) and offline (for tracing purposes) task execution monitoring in GPOS Linux.

1.3 Research method

Currently and in short term, I am focusing on the implementation of HSF. Looking at the HSF implementation, the research method used so far is the empirical method. Measuring the timing parameters such as the overhead time of the implementations and the schedulers, response time of tasks, input/output jitters and so on. Such measurements are important for comparison purposes (against other HSF’s, flat schedulers etc.) and for scheduling analysis.

The execution monitoring project also relates to time measurements, although it is the application’s objective to do so.

To enable logical resource sharing between tasks in HSF, a proper synchronization protocol should be used. Many synchronization protocols are proposed (such as HSRP [8] and SIRAP [7]) but none of them are implemented in a Real Time Operating System (RTOS). It is very interesting to implement these protocols and compare them in between, in terms of the overhead produced by the protocols. My on-going work is to implement and compare/evaluate these synchronization protocols based on the time measurements.

For scheduling analysis with included scheduling overhead, time measurement of the scheduler is important. But also to model the scheduler in terms of mathematical expressions and to incorporate these in existing scheduling analysis.

Looking at the time efficiency of the implementation, algorithms are measured/compared, for example queue handling algorithms. The purpose is to find the most time effective algorithms.
2 Current research issues and hot topics

The current front of research in hierarchical scheduling is first of all how to synchronize global resource sharing among different real-time applications [7, 8, 9]. The major problem with synchronization in hierarchical systems is how to deal with the situation when a task has a global resource locked and the application’s (for which the task belongs to) allocated CPU budget expires.

Another hot topic is of course multi-core in general, this area is relative unexploited. The work done so far (that is of interest) includes scheduling in the case when there is one task queue shared by all cores (global scheduling). Scheduling can also be done with one queue per core, today’s state of the art scheduling is applicable in this case but it minimizes utilization of all the cores. With global scheduling, an algorithm known as Pfair [10] has shown to be optimal but it has a very large overhead in practical use.

Areas that are unexploited or almost unexploited are for example hierarchical scheduling as well as synchronization of shared resources in multi-core.

3 Central literature

Hierarchical real-time scheduling is indeed part of the research area real-time systems. Relevant to hierarchical scheduling are the periodic task model, rate monotonic algorithm and EDF scheduling [1]. This paper is the basis of most of today’s real-time system theory.

The key point in hierarchical scheduling is how to model it and perform scheduling analysis. The first effort to model the CPU resource (as partitions) was done in 2001 [5], the model was called bounded-delay resource partition model. This model is non-periodic.

The resource model we use is called the periodic resource model [2] which is (as the name implies) periodic. The reason for choosing this model is it’s property of separating local and global scheduling analysis, the applications do not need to have knowledge about global scheduling nor other applications. This paper is maybe not a seminal paper but it has great influence in our research. The synchronization protocol SIRAP [7] is for example based on the periodic resource model as well as the analysis for HSF [6].
4 Key conferences and leading research groups

The conferences of my interest are those that are tightly connected to real-time systems. After some searching (and discussions with my supervisors) I have come up with a list of key conferences, listed in the order of importance. Finding the most attractive research groups in the area of real-time systems was done by searching the net. Due to my visit to Holland and RTSS'08 I have come in contact with some of these groups.

4.1 Key conferences

• **Real-Time Systems Symposium (RTSS)**
  RTSS is the conference of conferences in real-time computing, including tracks in real-time operating systems and scheduling.

• **Euromicro Conference on Real-Time Systems (ECRTS)**
  ECRTS is another conference in the area of real-time computing, including tracks in scheduling and schedulability analysis.

• **IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)**
  RTAS is a conference in the area of embedded and real-time systems and computing, including tracks in real-time resource management and scheduling.

• **IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)**
  RTCSA conference is focused on real-time and embedded systems, tracks in operating systems & middleware and quality of service & scheduling are included.

• **International Conference On Embedded Software (EMSOFT)**
  EMSOFT is a conference for research in embedded systems software.

• **IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)**
  ETFA is a conference for factory automation and emerging technologies in industrial automation with a track in real-time and (networked) embedded systems.
4.2 Leading research groups

- **Real-Time Systems Research Group at York**
  This group (stationed in York, England) is one of the largest academic research groups on engineering of real-time systems. They have conducted a lot of work in hierarchical scheduling. For example hierarchical scheduling analysis [11] (which has a different approach than Shin and Lee [2]) and synchronization of global shared resources in hierarchical systems (HSRP) [8].

- **System Architecture and Networking (SAN) at Eindhoven University of Technology**
  This group (stationed in Eindhoven, Holland) have a subdivision that works with hierarchical scheduling for soft real-time systems (multimedia products). The head of this section (Reinder Bril) and his students collaborate with our group.

- **Real-Time Systems Laboratory (RETIS lab) at Scuola Superiore Sant’Anna University**
  This research group (stationed in Pisa, Italy) conducts research in real-time embedded systems design and analysis. Much work has been done in hierarchical scheduling in this group including practical implementations of such systems which matches our research quite well.

- **Real-Time Systems group at The University of North Carolina**
  This group (stationed at Chapel Hill, USA) conduct a lot of research in hierarchical scheduling, synchronization and multi-core.

5 Relation and relevance for own research

Looking at current research in this field, not much work has been done in multi-core with hierarchical scheduling and synchronization for which is very interesting for us. In this sense there is not much relevance in previous research compared with what we are planning to do.

Collaborations are being made with Department of Computer Science, Keio University in Tokyo Japan in the area of scheduling in Linux. Also, due to my visit in Holland, collaborations will be made with students from SAN (Eindhoven, Holland). Further, some initial collaborations have been made with people from Real-Time Systems Laboratory (RETIS lab), Pisa. The relevance to our research is in the area of hierarchical scheduling in Linux and µC/OS-II.
References


