Applications of wait/lock-free protocols to Real-Time Systems

Håkan Sundell

Computing Science
Chalmers University of Technology

Schedule

- ENEA OSE.
- Using information available in RT in the design of WF/LF protocols.
- NOBLE: A library of Non-blocking synchronization.

OSE

- A Swedish Commercial Real-Time Operating System , by Enea in Täby.
- Developed with portability in mind
- Written in C and Assembler
  - Portable kernel in C consists of 30000 lines of sourcecode. Mainly constructed by one author.
  - Separate targetinterface for each platform, written in C/Assembler.

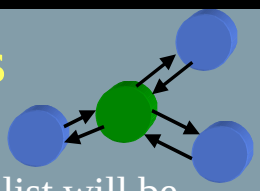
OSE concepts

- Limited API
- No Shared Memory
- Limited dynamic memory allocation.
- Fixed priority scheduler
- Interrupt re-prioritizer
- Uniprocessor system

OSE communication

- Task to Task
  - Signal oriented
  - Send/Recieve kernel calls
  - Messages can be only multiples of 2 in length, 64, 128, 256 and etceteras
- System to System
  - An add-on called Linkhandler can handle signals between tasks on different systems

OSE limitations



- Implementing a shared queue or list will be unefficient.
  - Has to be implemented using messages, two-ways.
- Shared data structures are common in system design.
- Have to transform the system to the signal model to use in OSE.

OSE project

- Developing test platform
  - Writing target interface for x86.
  - Developing using MS Visual C++ compiler and environment.
  - Running under DPMI32 + Win32 emulator with PE-loader, runs fully with preemption under DOS and partly under Win32.
- Extending and modifying the OSE kernel

Using information

- Many lock- and wait-free algorithms are theoretical oriented and general.
- Some are unefficient or even impossible to implement practically: use of unbounded memory.
- Often exhaustive usage of ”heavy”-atomic synchronization primitives.

Using information

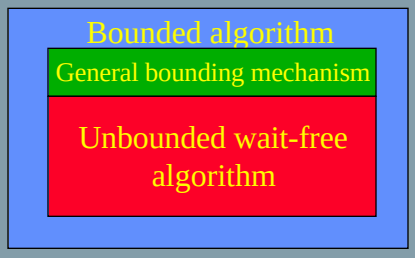
- Solution: Exploit the information that is available in the actual system
- Several approaches
  - Quantum based scheduling (Anderson et al.)
  - Priority based scheduling (Anderson et al.)
  - Timing information (Chen and Burns et al.)
    - We have extented this approach

Using information

- Our contribution (Timing Information from Real-Time Systems)
  - Wait-free Snapshot with bounded memory requirements and no atomic primitives.
    - Experiments show significant speed improvements.
  - Wait-free Shared buffer with bounded memory requirements and no atomic primitives.
    - Examples show significant improvements.

Using information project

- Generalising methods
  - Applying timing information to bound wait-free algorithms requiring time-stamps.



NOBLE: a library of non-blocking synchronization

- A library of of lock and wait-free algorithms for shared data structures.
  - Queues
  - Lists
  - Buffers
  - Snapshots

NOBLE: a library of non-blocking synchronization

- Provides a collection of data types and algorithms in a form which allows them to be used by non-experts.
- Gives a precise and readable specification for each of the data types and algorithms.
- Contains efficient implementations for each of the data types.

NOBLE: a library of non-blocking synchronization

- Multiprocessor support
  - Sun Enterprise 10000 (Sparc)
  - SGI Origin 2000 (Mips)
- Written in C
- Tested and evaluated

NOBLE: a library of non-blocking synchronization , project

- Extend to embedded systems
  - Common uni- and multi-processor systems
  - 8-bit processors with/without or different support for atomic synchronization primitives.
  - Asterix