CSC 714
Real Time Computer Systems

Project Proposal

Implementation of EDF, PIP, PCEP in BrickOS

Sushil Pai
spai@ncsu.edu

Project URL: http://www4.ncsu.edu/~spai/csc714

Objective:
Add support for EDF (Earliest Deadline First), PIP (Priority Inheritance Protocol) and PCEP (Priority Ceiling Emulation Protocol) in Brick OS. Currently Brick OS supports only static priority scheduling which does not perform any kind of deadline monitoring and resource management.

Introduction:
BrickOS (previously known as LegOS) is an open source embedded operating system, featuring preemptive multitasking, dynamic memory management and IR networking. It is designed to run on a Lego Mindstorm RCX brick based on the Hitachi H8/3292 microcontroller. It was started by Markus Noga in October 1998. The default scheduler that BrickOS supports is Static priority preemptive scheduling. This scheduling does not perform any kind of deadline monitoring and resource scheduling.

The Earliest Deadline First (EDF) algorithm is a dynamic priority-scheduling algorithm in which the priorities of individual jobs are based on their absolute deadlines. An EDF algorithm can generate a feasible schedule for a system of N independent, pre-emptable tasks as long as the total density of the system is less than 1. Hence EDF is an optimal scheduling algorithm.

In the priority inheritance protocol (PIP), the resource holder inherits the priority of the highest priority blocked process. When a thread tries to lock a resource using this protocol and is blocked, the resource owner temporarily receives the blocked thread's priority, if that priority is higher than the owner's. It recovers its original priority when it unlocks the resource.

Priority Ceiling means that while a process owns the resource lock it runs at a priority higher than any other process that may acquire the resource. In the priority ceiling solution each shared resource is initialized to a priority ceiling. Whenever a process locks this resource, the priority of the process is raised to the priority ceiling. This works as long as the priority ceiling is greater than the priorities of any process that may lock the resource.

Implementing a scheduler based on EDF/PIP/PCEP in the scheduler would take care of the above mentioned problems (deadline monitoring and resource management). The only drawback is that there would be no support for dynamic creation of tasks (which is provided by BrickOS using the command execi).

LNP stands for LegOS Network Protocol. It allows for communication between brickOS-powered RCX, and host computers. This is required for verification of the scheduler operation.

Challenges:

- Understanding the current implementation of Brick OS kernel (in particular the Scheduler)
- Understanding the H8/300 instruction set as parts of the scheduler are often implemented in assembly language.
- Verification of the functionality of PCEP (using LNP based communication between the RCX and a PC)

Outline for the solution:

- Familiarize with the existing scheduling mechanism of Brick OS
- Identifying the data-structures and functions associated with tasks and resources that need to be changed for implementing the scheduling mechanisms
- Implement communication between the RCX and a PC using LNP (for verification of the scheduler)
- Implementing PCEP
- Verification of results for PCEP using LNP (the output of the scheduler will be provided to the host computer over IR)
- Implementing PIP
- Verification of results for PIP using LNP
- Implementing EDF
- Verification of results for EDF using LNP
- Implementing PCEP with EDF
- Verification of results for PCEP with EDF using LNP
- Implementing PIP with EDF
- Verification of results for PIP with EDF using LNP

References:

- BrickOS Kernel Documentation
  http://legos.sourceforge.net/docs/kerneldoc.pdf
- LegOS Documentation
  http://user.it.uu.se/~tobiasa/RT-TF01/legosdoc/index.html
- Hitachi 8/300 Processor architecture and instruction set
  http://moss.csc.ncsu.edu/~mueller/rt/mindstorm/h3314.pdf
- LegOS Network Protocol
  http://legos.sourceforge.net/files/linux/LNPD/