CSC 714
Real time computer systems
Project Proposal

Implementation of EDF, PIP, and PCEP in RT-Linux

Amit Choudhary (achoudh4@unity.ncsu.edu)
Nitin Shrivastav (nshriva@unity.ncsu.edu)
Ramnath Venugopalan (rvenugo@unity.ncsu.edu)

Problem Description:

Our objective is to add support for Earliest Deadline First, PIP and PCEP in RT-Linux. The default scheduler in RT-Linux is a fixed-priority scheduler. There is no deadline monitoring or resource management (e.g. support for priority inheritance) functionality with the distribution we obtained from ftp.fsmlabs.com.

Background:

Real-time support in Linux: In Linux, the kernel code is non-preemptable in critical-sections. So, the interrupts generated during this time will be lost. Hence *hard* real-time guarantees are not possible. [10]

RT-Linux background: RT-Linux was developed at the New Mexico institute of technology, Socorro NM, by Victor Yodaiken and Michael Barabanov. RT-Linux is an extension to Linux OS that has been designed to handle time-critical tasks. RT-Linux runs the Linux OS as its lowest priority task. Linux is made completely preemptable and if CPU has to be dedicated to more important real-time tasks, then Linux can be completely disabled. In RT-Linux, the real-time kernel handles all the interrupts and then passes those interrupts to the Linux kernel that are generated for it, only if there are no real-time tasks to run. Real-time tasks cannot use Linux system calls or communicate directly with Linux processes as this would lead to unbounded timing constraints. Real-time tasks can use shared memory and FIFO queues as IPC mechanisms. [8][9]

Challenges:

1) Understanding the current implementation of RT-Linux kernel
2) Verification of the functionality of EDF, PCEP, PIP

Outline for the solution:

Our approach to the problem is the following:

1) Installation of RT-linux.
2) Familiarizing ourselves with the RT-linux kernel and its scheduling mechanism.
3) Identifying the data-structures and functions to be changed for implementing the scheduling mechanisms.
4) Implementing EDF.
5) Verification of results
6) Implementing PCEP & PIP.
7) Verification of results.

References:

1. http://www.linuxdevices.com/articles/AT5698775833.html
2. http://www.linuxdevices.com/articles/AT7168794919.html
3. www.fsmlabs.com
4. http://starship.python.net/crew/gandalf/rtlinux/luz.cs.nmt.edu/%257Ertlinux/whitepaper/short.html
5. http://216.239.39.120/translate_c?hl=en&u=http://www.linuxfocus.org/English/May1998/article44.html
6. http://216.239.39.120/translate_c?hl=en&u=http://bernia.disca.upv.es/~iripoll/publicaciones/research.html
7. http://www.rtlinux.at/documents/documentation/RTLManual/RTLManual-1.html
8. http://www.fsmlabs.com/developers/man_pages/
9. http://www.fsmlabs.com/developers/white_papers/
10. http://alpha400.ee.unsw.edu.au/elec2042/rtlinux/rtlinux.pdf

Project Page Link:

The project page can reached at: http://www4.ncsu.edu/~achoudh4