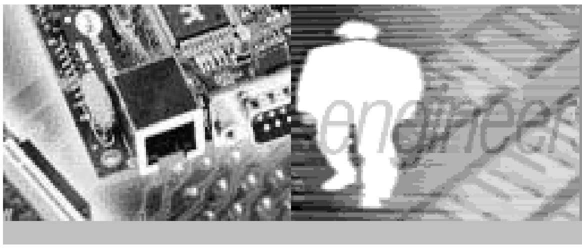
| Introduction to Real Time Linux |

>> A Practical Approach from the Real Time Linux Kernel Developer's perspective

Course Outline

Norik Systems

www.norik.com



| Real Time Linux Jumpstart |

• Overview

While market analysts and others focusing on the business side of the computing industry have become aware of the growing importance of Linux, a secondary market exists with potentially just as much impact: real-time extensions for Linux. Indeed, engineers designing embedded systems have come to embrace Linux as a genuine alternative to more traditional real-time operating systems.

This 5-day *Introduction of Real-Time Linux Training Course* is designed to examine what is involved in working with real-time Linux and is based on the experiences of someone who has devoted many years working on Real-Time Linux based systems. This course will cover topics like installation of base system from scratch (based on a current Distribution), patching for RTLinux and reconfiguring the kernel, compiling, installing the new kernel and modules, and configuring and installing RTLinux modules with examples.

At the end of the course, the participant will learn:

- **To install RTLinux from scratch on X86 UP and SMP**
- **Benchmark an existing hardware system and interpret the results**
- **Design and write simple RTLinux multithreaded applications**
- **Understand and comprehend the basics of RT-driver writing**

• Participants

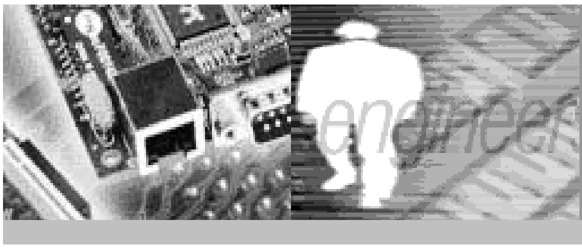
Students are software developers building embedded systems, project managers looking to know more about Real Time Embedded Linux and engineers who want to have a hands-on feel about Real Time Embedded Linux.

• Prerequisites

- Students are expected to be proficient in the C programming language.
- Well-versed in operating system concepts and familiar with Linux or a comparable development system is essential.

• Training Approach

- Hands-on
  There are continuous, ongoing hands-on throughout the course. Target boards will be provided.
- Small class size
  Not more than 15 students.
- Up to date information.
  We update the materials before every event.
- Straightforward explanations
  Technical concepts and terms are explained in English. You will walk away with a thorough understanding of the tradeoffs of using Embedded Linux in embedded systems and applications.
- 4 weeks of post e-mail support for course related questions on the course mailing list



Certification

There will be daily home-work assignments. The knowledge is tested on the last day with a hands-on session. Certification of successful participation is only granted if a minimum program is achieved and at least two assignments are completed.

Course Concepts

- Primarily hands on - main focus on X86 UP with introduction of SMP specifics.
- Clear focus on the power of the command line, introducing powerful Unix/Linux tools along the way.
- Error Analysis on the system level.
- Project management issues are also taken into account - tools used - environment policies.
- Using CVS and participating in community efforts

Trainer

Nicholas McGuire is an Embedded/Real-Time Linux Software developer from Austria. He is also the official Real-Time Linux core designer and maintainer of RTLinux/GPL. He has 11 years experience in embedded real-time Linux applications with high performance computing and cluster administration. He has given talks, seminars and workshops for Real-Time Linux and Embedded Linux in Austria, Germany, France, Italy, Sweden, Spain, UK, Slovenia and the USA since 2000. He co-founded and organized The Real Time Linux Workshop in Vienna 1999, Orlando 2000, Milan 2001, Boston 2002, Valencia 2003, Singapore 2004 and Lille 2005.

For more information, please see Trainer’s profile attached.

Course Fees

Please call to inquire.

Duration

5 full working days – 9:30am till 5:30pm

- **Course Outline**

  - How does Real Time Linux work
    - Interrupt interception
    - Interrupt emulation
    - GPOS in the idle task

  - RTLinux API v3 and the compatibility notes

  - A simple kernel module
    - Mandatory init module
    - Cleanup module
    - Real-time threads
    - Real-time handlers

  - Introduction to RTLinux API v3

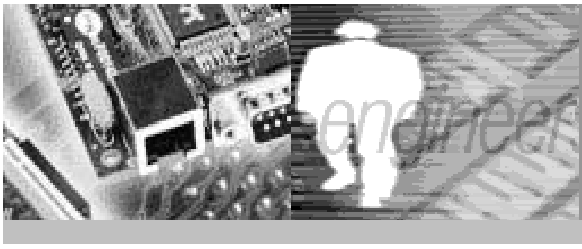
  - RTLinux core kernel modules
    - How to run them
    - Passing module parameters
    - Kernel facilities available
    - To access in RT-context
    - kthread, tasklets, char-dev's
    - Examples

  - Extensions to RTLinux
    - Using RT serial communication
    - Using RT Socket API
    - Using light weight IP stack for RTLinux

  - System Evaluation and Debugging
    - Regression tests and performance measurements
    - Interpretation of results
    - Using of debugger rtl_debug
    - Using of tracer rtl_tracer
    - What to do with kernel oops

  - Introduction of POSIX API
    - Thread creation and attributes
    - Thread termination
    - Synchronization primitives
    - Cancellation points and cleanup handlers

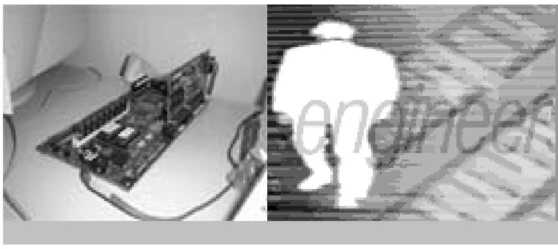
  - Designing an RT-Task
    - How to split the job
    - Outlining the code for a simple driver
    - Passing arguments at module initialization and the proc interface
    - Using Interprocess communication initialization (SHM, FIFO's, signals)
    - Process synchronization
    - Resource management



- **Course Outline**

  - Basics of driver writing
    - Basics of Linux Kernel Modules
    - Outlining the structure of a RTLinux driver
    - Using Linux kernel facilities
    - Communicating with the non-RT task
    - Using Linux logging facilities

  - A Sample Driver
    - init_module
    - Driver “library” module
    - Cleanup module
    - Integrating the driver with you



- **For more information**

If you have any further enquiries on our Embedded Linux courses, do not hesitate to contact us.

Norik Sistemi d.o.o.
Stari trg 9
SI-3210 Slov. Konjice
Slovenia, Europe

Tel: +386 41 540 545
Tel: +386 3 759 3077
Fax: +386 3 759 3078
Email: **info@norik.com**