

# RTLinux 课程介绍

## 介绍

市场分析和实际的情况证明了 Linux 正日益成为计算机工业的重要方面,尤其是在嵌入式系统: Linux 正在成为一个重要的操作系统,扩展 Linux 实时性的问题也突显出来。事实上,设计嵌入式系统的工程师已经将 Linux 作为越来越多的传统实时性操作系统的替代。五天的 RTLinux 培训课程是专门为基于 Linux 的实时性工作和嵌入式 Linux 开发工程师和项目经理而专门设计。课程设计别具一格,重视实际的使用,通过一周的时间,学员可以使用开源的 RTLinux/GPL 已开发和构建一个实时的 Linux 系统以及调试工具。

## 参加者

嵌入式系统的软件开发人员,想了解更多实时性嵌入式Linux的项目管理人员,和想更深入学习实时性嵌入式Linux的工程师。

## 要求

- 精通C语言编程。
- 精通操作系统概念,熟悉Linux或者相似的开发系统。

## 培训方法和课程设置

### ○ 实验

在课程中有不断的实验,一般情况是每个人使用一台X86/PC。

### ○ 班的大小

原则上最多有20人参加,这个限制是考虑到课程培训的互动交流,对每一个学生的高度关注。

### ○ 教材更新情况

课程的教材不断更新和保持最新的。

### ○ 说明

技术概念和定义用英语解释和必要的中文翻译。最后离开的时候,你能够完全掌握在嵌入式系统中如何使用Linux和设计应用程序,培训方式是让所有的参加者回家以后,能够从最原始安装系统,工作,解决实际的问题,不是仅仅提供带原码的软件,文档,而是保证所的过程清晰明了。重心是掌握实际的操作技能,而不是仅仅解释理论,因此我们主要介

这些非常重要工具和概念背后的核心技术。一个星期的培训对于参加者应该足够知道如何进行实际的项目和获得经验。要注意的是，我们没有使用特殊的硬件来保证这些经验,我主要应用典型的 PC系统,基于这次培训的经验,大多数工具的安装和使用没有问题,也很容易的移植到实际的环境上..

o 课程培训完成后四个星期以内老师可以通过电子邮件支持回答与培训相关的问题

## 认证

每天都有作业。要求掌握的知识会在最后的实验中检验。要通过成功的认证，必须完成两个的编程任务。最后认证通过将发放由Nicholas Mc Guire教授签名的证书。

## 培训师

培训师来自于奥地利的 RTLinux/GPL 内核的设计者和维护者，嵌入式 Linux 专家，著名 Nicholas Mc Guire 教授。Nicholas 有十几年的高性能计算和嵌入式/RTLinux 应用经验,在 2000 - 2003 年曾经是 FSMLabs 德国公司 CEO 和高级工程师，负责 RTLinux/GPL 的开发，维护，咨询和国内的培训课程。2003 年 Nicholas 创建 OpenTech EDV-Research GmbH 承揽欧洲西门 CT, 医疗和自动化部门包括 RTLinux 和通讯系统在内的各种项目。2005 年 7 月 Nicholas 正式入兰州大学信息学院和李廉教授，周庆国博士组建了分布式和嵌入式系统实验室（DSLAb

Nicholas 以 Der herr Hofrat 的名字活跃在开源社区，Nicholas 还是 Real Time Lin Foundation 的创始成员，自 2000 年至今成功组织了 8 届 RTLinux 国际研讨会。

Nicholas在世界各地演讲和在新加坡，德国，兰州大学开设 RTLinux/GPL，嵌入式 Linux调

嵌入式 Linux和开源项目管理等课程。

兰州大学分布式和嵌入式系统实验室的学生和麦克泰公司的工程师参加助教和辅导。

## 课程费用

电话资讯北京麦克泰软件公司北京办公室，86 - 10 - 62975900，电子邮件 info@bmrtech.c

## 时间

4 月 23 日 - 27 日，9:00 –12:00am 到 1:00-6:00pm，具体联络麦克泰公司

## RTLinux 课程内容

根据参加者的掌握的快慢稍微变化，一般情况大多数题目会被涵盖，在主要的介绍内容被仿的情况下，一些特殊的题目会跳过或者减少。标记（）符号的内容是为非常快的小组准备的？Installation of Linux/RTLinux:

–Installing the base system - Slackware 10.2

- \*basic system setup from scratch
- \*basic configuration of networking
- \*security issues
- Preparing for RTLinux
  - \*patching the kernel sources
  - \*configuring the kernel to fit your system
  - \*configuring kernel for RTLinux/GPL
  - \*installing the RTLinux executables and kernel modules
- ? Basic filesystem:
  - Building a filesystem from scratch
    - \*what is a filesystem ?
    - \*The naive filesystem by copying files
    - \*testing the filesystem with chroot
    - \*limitations of this approach
  - minimum filesystem with UML
    - \*building a real minimum filesystem from source
    - \*configuring the UML kernel
    - \*booting a minimum filesystem under UML
    - \*fixing problems
    - \*() networking in UML
- ? Basic RTLinux concepts:
  - How does RTLinux work
    - \*interrupt interception
    - \*what happens to Linux GPOS
    - \*communication
    - \*limitations and outlook towards XtratuM
  - The RTLinux API (V3) and compatibility notes
  - Introduction to the RTLinux API
    - \*pure POSIX
    - \*interrupt management
    - \*IPC
    - \*Communication
    - \*Interacting with the Linux kernel
    - \*Accessing I/O ports
- ? Kernel Modules:
  - Basics of Linux Kernel Modules
  - A simple kernel module
    - \*hello world in kernel space
    - \*symbol management
    - \*oops -ing and decoding an oops
  - What do the RTLinux Modules provide
  - Passing arguments at module initialization
  - the proc interface

- Using Interprocess communication facilities (SHM,FIFO ' s,signals)
- ? Modules and Examples:
  - The RTLinux kernel modules
  - how to run the examples
    - \*passing parameters
    - \*rebuilding cleanly
    - \*limitations of examples
  - Using the debugger and tracer
    - \*configuring for debugging
    - \*debugging with gdb via gdb - stubs in rt - context
    - \*concept of the RTLinux tracer
    - \*configuring for tracing
    - \*tracing an example and analyzing results
  - Initial benchmarking
    - \*why do it ?
    - \*Interpretation of basic benchmark results
    - \*(\*) tuning the system
  - platform evaluation issues
    - \*comparing environments
    - \*documentation of configuration
    - \*extended tools available
- ? Designing an RT - Task:
  - Basic Design concepts of RTLinux
    - \*How to split the job
    - \*Communication basics
    - \*when NOT to use RTLinux/GPL
  - Interprocess communication
    - \*fifos
    - \*shared memory
    - \*having " fun " with signals
- ? Basic Driver concepts:
  - Defining the Interface
  - Outlining the structure of a RTLinux driver
  - Splitting your rt - application
  - Using Linux kernel facilities
  - Communicating with the non - RT task.
  - Driver Control and Monitoring from the Linux side
  - Using Linux logging facilities.
- ? () A Data - acquisition driver line by line:
  - init module
  - The driver " library " - module
  - Accessing the driver functions
  - cleanup module

- the Makefile
- Integrating the driver with your rt-task
- license issues

## 培训的资料

符合 GPL 的 Linux 发行版的 CD，包含所有的系统软件和补丁。  
培训软件的 CD，包含所有的例子，和软件工具。  
硬拷贝的培训教材。

## 培训组织者介绍

北京麦克泰软件技术有限公司（BMR）成立于 1995 年，是国内首家从事嵌入式软件专业公司。总部位于北京海淀上地，上海和深圳分别设有办事处和分公司，为客户提供本地的技术和商支持。BMR 在中国推广业界最知名的嵌入式系统和电子设计品牌产品。产品涉及嵌入式操作系统，开发工具，嵌入式平台，电子设计工具和培训服务等。BMR 获得《电子产品世界》组织选 2004-2005 中国本土嵌入式十佳企业 之一，BMR 是中国软件行业协会，嵌入式分会和开联盟的会员。BMR 长期致力开源软件在嵌入式和实时系统中的技术推广和应用工作。  
关心麦克泰公司更多的产品和服务 请访问 [www.bmrtech.com](http://www.bmrtech.com)