Training Catalogue
29/05/2020

KAÏNA-COM
TRAINING CATALOGUE

Real Time and Embedded Linux Software Development

hands-on course focuses on real time and embedded Linux with real time and embedded aspects of kernel programming. Lab work using an embedded device, is an integral part of the course.

Nos locaux
KAÏNA-COM France
LE CARRÉ HAUSSMANN II
6 Allée de la Connaissance
77 127 Lieusaint

Contact
+33(0)9 50 20 91 64

E-mail
info@kaina-com.fr

Site Internet
www.kaina-com.fr

KLI001 – Real Time and Embedded Linux Software Development

Reference

KLI001

Experience

- ☐ Beginner
- ☒ Intermediate
- ☐ Advanced

Duration

Training Program:

- 5 days

Training Method

- ☐ I: i-learning, individual training (web-based training)
- ☒ V: v-learning, virtual class
- ☐ C: c-learning, classroom training

KAÏNA-COM
LE CARRÉ HAUSSMANN II,
6 Allée de la Connaissance
77127 Lieusaint - France

Price

2.526,00 € HT

Prerequisite

Linux Basics, Linux Introduction or equivalent. Linux Systems Programming or equivalent.
Knowledge of C. Basic knowledge of device drivers and kernel modules is essential.

Audience

Embedded and RT programmers developing devices using the Linux kernel and driver developers for internal or external peripherals

Continued on next page

KLI001 – Real Time and Embedded Linux Software Development, Continued

Objective

The GNU / Linux operating system is the operating system of choice for many embedded and real time developers: the main reasons being that the source code is free, there are no runtime royalties and it is a robust reliable operating system with excellent networking support. This hands-on course focuses on real time and embedded Linux with real time and embedded aspects of kernel programming. Lab work using an embedded device, is an integral part of the course.

Continued on next page

KLI001 – Real Time and Embedded Linux Software Development, Continued

Course Contents

Course Contents :

Table 1: KLI001 - Course Contents

Chapter	Description
Introduction	• Linux overview • Real time and embedded • The kernel and its role • Linux supported architectures
Build Root	• Project overview • Getting buildroot • Quick start • Configuration interfaces • Using a predefined configuration
Cross tool chains	• The need for cross tool chains • Tools naming convention • Getting and installing a tools chain • Cross building software • Cross debugging • uClibc
Configuring and Building the Linux Kernel	• Getting the sources • The structure of source tree • Configuring and building the kernel • Compiling the kernel • Kernel modules: – Cross compiling modules – Integrating modules into the kernel source tree • Configuring buildroot: – Configuring the kernel in buildroot

Continued on next page

KLI001 – Real Time and Embedded Linux Software Development, Continued

Course Contents, continued

Chapter	Description
Customizing Buildroot	• Integrating Additional packages into buildroot: – dl – packages – config.in • Overlays
Device Trees	• Working without device trees • What is a device tree • DTS and DTB • Device tree integration into driver code. • The syntax of DTS files
Linux Boot Sequence	• Embedded Linux boot process • Kernel boot parameters • Bootloaders, U-Boot • Buildroot configuring system components: – init, busy box, U-Boot • root-fs: – initrd & initramfs – overlayfs
Net-Booting and The Network File System (NFS)	• How does NFS aid the embedded development process • Preparing NFS • Mounting an NFS volume • NFS daemons • Exports file • root-fs over NFS • tftp • DHCP

Continued on next page

KLI001 – Real Time and Embedded Linux Software Development, Continued

Course Contents, continued

Chapter	Description
User-mode Programming	• librt overview • Scheduling policies and priorities. • CPU affinity • Memory • RT signals: – explanation – comparison with standard signals • Asynchronous I/O • POSIX IPC: – Semaphores – Message Queues – Shared memory • POSIX timers. • Tips for improving user space RT performance. • Command line tools for manipulating scheduling policy / priority, and CPU affinity

Continued on next page

KLI001 – Real Time and Embedded Linux Software Development, Continued

Course Contents, continued

Chapter	Description
Linux and Real Time	• RTOS memory issues and Linux. • Linux hardware interaction • Latency (kernel, interrupt, scheduler) • Kernel preemption • Linux hard real time extensions • Applying the RT patch • Threaded IRQ's • Voluntarily giving up CPU – cond_resched • Controlling kernel preemption: – preempt_disable – preempt_enable – preempt_count • spinlocks and raw spinlocks. • Priority inheritance • Priority inversion • Don't do's
Introduction to The Yocto Project	• Project overview • The Yocto project development environment • Setting up a Yocto project: • Supported build hosts • Build host packages • Getting Yocto • Example – Building an image and testing it on an emulator • Development models: – System development – Application development • Image development: – Toaster – Hob • Receipes

Continued on next page

KLI001 – Real Time and Embedded Linux Software Development, Continued

Course Contents, continued

Chapter	Description
The End	• Summary • Q&A • Evaluation