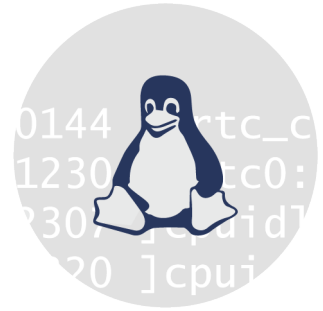


Embedded Linux in Theorie und Praxis



Beschreibung des Seminars

Der Einstieg in ein so mächtiges Werkzeug wie Linux ist nicht trivial! Ziel des Seminars ist es, Ihnen grundlegende Embedded Linux Konzepte sowie die Handhabung von Linux zu vermitteln.

Sie erfahren, welche Vor- und Nachteile Linux bietet und welche Komponenten man braucht, um ein Embedded GNU / Linux System zu bauen. Des Weiteren wissen Sie am Ende des Seminars, woher man entsprechende Komponenten beziehen kann und wie man diese konfiguriert, übersetzt und installiert. Sie lernen außerdem, woher man bei Unsicherheiten Hilfestellungen erhalten kann und welche Lizenzmodelle im Linux-Umfeld angewandt werden.

Mit Hilfe von zahlreichen Hands-on-Beispielen erlernen Sie, wie man ein Embedded GNU / Linux System aus Mainline-Komponenten zusammenstellt.

Zielgruppe

Projektmanager, Software-, Hardware-, Entwicklungs- und Systemingenieure

Voraussetzungen

Grundkenntnisse zur Bedienung von Linux (Ubuntu), Vertrautheit mit Embedded C Konzepten und Programmierung, die Fähigkeit, Software in C-Syntax zu entwickeln, und grundlegende Erfahrung mit Embedded Hardware (Eval Boards) sind von Vorteil.

Inhalte des Seminars

1. Tag: Grundlagen

- Einführung in GNU / Linux: Geschichte, Lizenzen, Standards, Arbeit mit Open Source, Spelunking, Unix Philosophie
- Eigenheiten von Embedded Linux:
 - Embedded Systeme
 - Gegenüberstellung Embedded Linux vs. Desktop Linux
 - Dysfunktionalitäten erkennen und beheben
 - Portierbarkeit
 - Für das Target bauen: Toolchains, C-Bibliotheken

- Eval Board - Beagle Bone Black
 - Booten (generisch) bzw. das Beagle Bone Black
 - SD-Karte partitionieren bzw. formatieren
 - Partitionen mit boot-loader, kernel, rootfs befüllen
 - serielle Konsole konfigurieren
 - Board mit GNU/Linux booten

2. Tag: Installieren und konfigurieren von Host und Target

- Toolkit installieren, NFS-Server, tftp-Server
- U-boot: auschecken, konfigurieren, cross-kompilieren, installieren
- Flattened device tree
- GNU/Linux kernel: ulmage, auschecken, konfigurieren, cross-kompilieren, installieren, Kernel Module
- Root File System
- Anpassungen: Dem Board Netzwerkunterstützung hinzufügen (U-boot Scripting, Netzwerkunterstützung in U-boot, maßgeschneiderter Kernel mit Netzwerkunterstützung)
- Rootfs über NFS
- Init (Sys-V, Upstart, Initng, Systemd), Bootgraph, Bootchart

3. Tag: Kernel Module, Treiber und Debugging im Überblick

- Kernel Module: Hello Kernel, module-init-tools, Kconfig, Kbuild, out of tree, in tree
- Device Treiber: Device Nodes, Character Treiber schreiben, Registrierung, Initialisierung, Miscellaneous Character Treiber
- Debugging / Profiling / Tracing im Überblick:
 - Einfache Debugging Werkzeuge: lsof, ltrace, strace, proc, top, netstat, syslog
 - Weitere Debugging Werkzeuge: gdb und target gdb, gdbserver, kgdb/kdb und agent-proxy, JTAG
 - Profiling: time, gprof, gcov, oprofile
 - Tracing: ftrace, kernelshark, LTTng
 - Verschiedene andere Werkzeuge: top, latencytop, powertop, powerdebug, crash, systemtap

Methodik und Seminarunterlagen

Vortrag und praktische Beispiele mit Host (Laptops mit Ubuntu 14.04.x LTS) und Zielsystem (z. B. Beagle Bone Black Rev. C - <http://beagleboard.org/BLACK>)

Die Geräte werden Ihnen während des Trainings zur Verfügung gestellt, jeweils ein Arbeitsplatz für zwei Teilnehmer.

Sie erhalten ein Arbeitsbuch (in Englisch), ein Beagle Bone Black Rev. C + Standard FTDI 3.3V to USB Kabel und nach dem Training einen Download Link mit Beispielen, um das Erlernete zu vertiefen.

Dauer

3 Tage

Preis

1.790 Euro zzgl. MwSt. pro Person.

Im Preis sind das Teilnahmezertifikat und die Seminarunterlagen sowie Snacks, Getränke und die Mittagessen in einem umliegenden Restaurant enthalten.

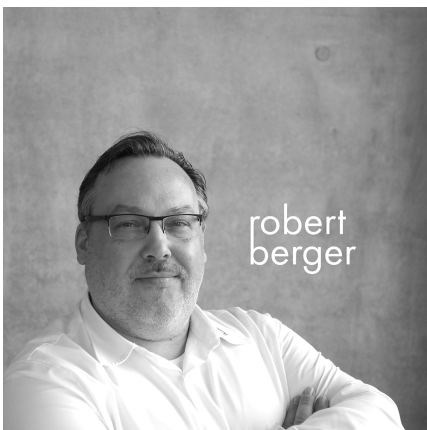
Teilnehmerzahl

6 bis 12 Teilnehmer

Veranstaltungsort

Technologiezentrum TechBase, Franz-Mayer-Straße 1, 93053 Regensburg

Referent



Robert Berger

hat seit 1993 Praxis- und Führungserfahrung in der Industrie bei Design und Entwicklung von Embedded Systemen. Er arbeitet seit Jahren mit GNU/Linux auch im Desktop/Server Bereich, aber hauptsächlich für "Eingebettete Systeme" (Automotive, Industrial Control, Robotics, Telecom, Consumer Electronics,...).

Seine Kerngebiete erstrecken sich von von kleinsten Echtzeit Systemen (FreeRTOS) bis hin zu Systemen mit mehreren Prozessoren/Cores und Embedded GNU / Linux (User-, Kernel-Space, Device Drivers, Hardware Interfacing, Debugging, Multi-core, Yocto Project) mit Schwerpunkt Free bzw. Open Source Software.

[Stand: März 2018]